QLectives – Socially Intelligent Systems for Quality
Project no. 231200

Instrument: Large-scale integrating project (IP)
Programme: FP7-ICT

Deliverable D2.1.1

Candidate theory models for cooperation algorithms

Submission date: 2010-03-01

Start date of project: 2009-03-01 Duration: 48 months

Organisation name of lead contractor for this deliverable: TUD

Project co-funded by the European Commission within the Seventh Framework Programme (2007-2013)		
Dissemination level		
PU	Public	X
PP	Restricted to other programme participants (including the Commission Services)	
RE	Restricted to a group specified by the consortium (including the Commission Services)	
CO	Confidental, only for members of the consortium (including the Commission Services)	

Document information

1.1 Author(s)

Author	Organisation	E-mail
David Hales	TUD	dave@davidhales.com
Nazareno Andrade	TUD	nazareno@gmail.com

1.2 Other contributors

Name	Organisation	E-mail
Tamás Vinkó	TUD	T.Vinko@tudelft.nl
Rameez Rahman	TUD	rrameez@gmail.com
Rahim Delaviz	TUD	rahim.delaviz@gmail.com
Dimitra Gkorou	TUD	d.gkorou@tudelft.nl
Johan Pouwelse	TUD	peer2peer@gmail.com
Nigel Gilbert	UniS	n.gilbert@surrey.ac.uk
Sergi Lozano	ETH Zurich	slozano@ethz.ch

1.3 Document history

Version#	Date	Change
V0.1	08-02-2010	First draft internal consortium version
V0.2	11-02-2010	Corrections and restructure - discuss user attention
V0.3	22-02-2010	Further corrections and additions
V1.0	01-03-2010	Approved version to be submitted to EU

1.4 Document data

Keyworlds	QLectives, peer-to-peer, evolution of cooperation, migration, group selection, indirect reciprocity, altruistic punishment
Editor address data	David Hales
Delivery date	01-03-10

1.5 Distribution list

Date	Issue	E-mail
	Consortium members	QLECTIVES@list.surrey.ac.uk
	Project officer	Jose.FERNANDEZ-VILLACANAS@ec.europa.eu
	EC archive	INFSO-ICT-231200@ec.europa.eu

QLectives Consortium

This document is part of a research project funded by the ICT Programme of the Commission of the European Communities as grant number ICT-2009-231200.

University of Surrey (Coordinator)
Department of Sociology/Centre for Research in Social Simulation
Guildford GU2 7XH
Surrey
United Kingdom
Contact person: Prof. Nigel Gilbert
E-mail: n.gilbert@surrey.ac.uk

University of Fribourg
Department of Physics
Fribourg 1700
Switzerland
Contact person: Prof. Yi-Cheng Zhang
E-mail: yi-cheng.zhang@unifr.ch

Technical University of Delft
Department of Software Technology
Delft, 2628 CN
Netherlands
Contact Person: Dr Johan Pouwelse
E-mail: j.a.pouwelse@tudelft.nl

University of Warsaw
Faculty of Psychology
Warsaw 00927
Poland
Contact Person: Prof. Andrzej Nowak
E-mail: nowak@fau.edu

ETH Zurich
Chair of Sociology, in particular Modelling and Simulation
Zurich, CH-8092
Switzerland
Contact person: Prof. Dirk Helbing
E-mail: dhelbing@ethz.ch

Centre National de la Recherche Scientifique, CNRS
Paris 75006,
France
Contact person: Dr. Camille ROTH
E-mail: camille.roth@polytechnique.edu

University of Szeged
MTA-SZTE Research Group on Artificial Intelligence
Szeged 6720, Hungary
Contact person: Dr Mark Jelasity
E-mail: jelasity@inf.u-szeged.hu

Institut für Rundfunktechnik GmbH
Munich 80939
Germany
Contact person: Dr. Christoph Dosch
E-mail: dosch@irt.de

QLectives introduction

QLectives is a project bringing together top social modelers, peer-to-peer engineers and physicists to design and deploy next generation self-organising socially intelligent information systems. The project aims to combine three recent trends within information systems:

- **Social networks** - in which people link to others over the Internet to gain value and facilitate collaboration
- **Peer production** - in which people collectively produce informational products and experiences without traditional hierarchies or market incentives
- **Peer-to-Peer systems** - in which software clients running on user machines distribute media and other information without a central server or administrative control

QLectives aims to bring these together to form Quality Collectives, i.e. functional decentralised communities that self-organise and self-maintain for the benefit of the people who comprise them. We aim to generate theory at the social level, design algorithms and deploy prototypes targeted towards two application domains:

- **QMedia** - an interactive peer-to-peer media distribution system (including live streaming), providing fully distributed social filtering and recommendation for quality
- **QScience** - a distributed platform for scientists allowing them to locate or form new communities and quality reviewing mechanisms, which are transparent and promote quality

The approach of the QLectives project is unique in that it brings together a highly inter-disciplinary team applied to specific real world problems. The project applies a scientific approach to research by formulating theories, applying them to real systems and then performing detailed measurements of system and user behaviour to validate or modify our theories if necessary. The two applications will be based on two existing user communities comprising several thousand people - so-called ”Living labs”, media sharing community tribler.org; and the scientific collaboration forum EconoPhysics.

Executive summary

The aim of this deliverable is to identify and translate theoretical models of cooperation formation into algorithms for ICT systems. We draw on the review of models from deliverable D1.1.1, and elsewhere, in the context of possible application areas in stream 4 (QMedia and QScience).

Translation of abstract cooperation models into ICT applications is a non-trival task. It is generally not the case that existing models can be directly applied without significant modification due to the engineering constraints that are inherent in deployable systems. With this in mind we aim for this deliverable to begin to define a framework which can be of general value for subsequent work within the workpackage and beyond. The main contributions of this deliverable are:

- Introduce some of the general issues in moving from abstract models to ICT applications. Here we make a distinction between the concept of a *user model* and a *protocol* - chapter 2.
- State two high-level QMedia application domains - chapter 2.
- List a set of candidate *user models* and *collective mechanisms* derived from the modelling literature - chapter 3.
- Discuss initial and on-going work in applying these models to the application domains - chapter 4.
- Propose potential future directions for this workpackage - chapter 5

Specifically, the high-level QMedia application domains are:

- Promoting seeding for media sharing. This involves providing incentives for peers to contribute upload bandwidth to others in the community.
- Promoting quality content and metadata. This involves providing tools and incentives such that users contribute high quality information (in the form of content and metadata) to the community.

The collective mechanisms selected to address the domains are:

- Indirect reciprocity
- Migration and group selection
- Altruistic punishment

We present an overview of existing work on indirect reciprocity and present initial directions of work for the latter two mechanisms.

Contents

Chapter 1

Introduction

The aim of this deliverable is to identify candidate theory models from stream 1 (as reported in QLectives deliverable D1.1.1), and elsewhere, that may be productively applied as cooperation formation algorithms for distributed ICT. In order to achieve this we have structured our work into a series of steps which we report on in this deliverable:

1. Identify areas for target applications that would benefit from increased cooperation.
2. Select appropriate user models which may capture the behaviour of users within such application areas.
3. Determine the possible collective mechanisms that can promote cooperation within the application areas for the given user models.
4. Design appropriate protocols, that could be deployed, that instantiate the collective mechanisms.

Step 1 is a comparatively straightforward task since it involves determining high-level requirements for given application areas that require high levels of cooperation between peers. Hence we do not need to specify how each requirement is to be achieved but rather what each broad requirement is. We describe our initial application requirements in chapter 2.

For step 2 we have a more complex task. Here we wish to select user behaviour models, informed by theoretical models (such as those discussed in deliverable D1.1.1) but plausible in the context of actual user behaviour in P2P systems. Actual user behaviour, in some domains, can be measured but it is often not the case that such measurements can be easily used to induce models that can be compared to theoretical models. This is due to several factors including the difficulty in separating the user model from the measurable outcome. For example, if one measures the bandwidth contribution of peers over time this is determined not just by user behaviour but also by external factors such as the network connection speed and the connectability of the user's computer (many users are behind firewalls or other limiting or filtering systems). However this is an open area of on-going research. In some sense the user models we select are hypotheses that can be tested first in simulation and then in deployed systems to determine if they produce expected outcomes. We describe candidate user models in chapter 3.

In step 3 we need to draw on existing theoretical models which propose collective mechanisms which may be applicable to the application areas we have selected and the user models we will consider. We need to consider those mechanisms that appear - as least at this stage - to be able to contribute to potentially deployable (and already deployed) P2P protocols. This requires that assumptions of the theoretical models can be considered to hold plausibly in real systems and that the important, often emergent, processes can be supported by the combination of user behaviour (as specified in the user models) and potential protocol implementations. We describe the selected candidate social mechanisms in chapter 3.

Step 4 is perhaps the most difficult step since it requires the design of protocols that apply the user models and collective mechanism models in a way that addresses the requirements identified in step 1. For this reason step 4. comprises two parts. The first is the use of simulation to test potential ideas for protocol designs and the second, the implementation of potential protocols within stream 4 of the QLectives project (comprising WP4.1-WP4.3). In order to realistically address this task we also identify on-going and developed protocols (for deployment in stream 4) and how they can benefit from and instantiate the collective mechanisms selected. We present our on-going work in these areas in chapter 4.

Chapter 2

Translating theoretical models into P2P applications

In this section we first discuss a number of issues that arise when moving from abstract theoretical models to application areas, specifically within peer-to-peer (P2P) distributed ICT systems. With these issues in mind we then present two high-level requirements for QMedia. QMedia aims to provide a media sharing client (based on Tribler) providing high quality media sharing via implementing fully decentralised downloading, video-on-demand, moderation and social collaboration tools. Finally we give a brief summary of the chapter.

2.1 Incentives and security

For a given application area cooperation may be interpreted in two distinct ways. We may be interested in promoting certain kinds of *user behaviour* or *software behaviour* or both. Since the P2P systems we are interested in are not centrally controlled this means we can not ensure users will behave in the way we wish. Also we can not be certain that software protocols (i.e. algorithms or so-called clients running on user machines) will not be changed by users (hacked) for nefarious purposes - either to increase some individual performance at the expense of the collective or just to bring down the system as part of a cyber-attack. Security in this context means interpreting cooperation as nodes in a P2P system running a correct version of the client. The ultimate security system would not allow clients to be hacked by somehow detecting and shutting out hacked variants of the client. The ultimate incentive system would mean the users were never rewarded (however that is defined, see user model section 3.1) for behaving in an anti-social way, however that is defined. An example from BitTorrent: there are many BitTorrent clients and some do not follow the standard and have been specifically hacked, such as BitTheif and BitTyrant - interestingly both these were done as academic experiments [26, 19] - this is a security focus. Many private file-sharing communities implement centralised enforcement mechanisms that punish peers who do not contribute sufficient bandwidth over time - this is a user incentive focus. Within the BitTorrent protocol the tit-for-tat (TFT) like file piece exchange (which the user does not need to know) provides incentives for users to share while they download - i.e. a user who limits their upload to a very small value should get low download speed [5] - this is both. These two ways of viewing cooperation are not completely distinct of course. One can view TFT as an incentive not to naively hack the client to not reciprocate.

2.2 User model and protocol

In abstract models it is almost always the case the behaviour of the agents (or nodes) is monolithic and generally is meant to represent people or perhaps the combination of people and algorithms. However, in the design of techno-social systems (such as P2P systems) it is necessary to clearly distinguish between the user model (what the user has control over and may influence) and the protocol (the algorithm that runs on the user machine that the user interacts with). For example, for an application that requires no input from the user, all the strategy would be coded in the protocol. Conversely if we were just modelling people interacting without mediating algorithms then all the strategy would be considered as the user model. Any real P2P application is some combination of the two since minimally a user needs to keep running an application for it to function (assuming the user controls what runs on his or her machine). Abstract cooperation models can supply ideas for both user models and protocols (algorithms) but in moving such models to ideas for applications it is essential to clearly specify the interpretation. This effects how we design / test incentives. One way to think of this is the difference between user incentives and protocol incentives. For user incentives we ask "what kinds of mechanisms encourage users to behave in a socially beneficial (cooperative) way"; for protocol incentives we ask "what kinds of mechanisms encourage software to behave in a socially beneficial (cooperative) way". Figure 2.1 illustrates how user models and protocols relate within a distributed P2P system. Note the possibility of both in-protocol communication (between users through the protocol) and out-protocol communication (between users via other means) in addition to protocol-to-protocol communication.

Aspects of a user model can be incorporated into a protocol if it can be sufficiently well specified as to be implemented algorithmically. For example, the TFT mechanism can be considered as a form of user model [2] which is sufficiently well defined in the context of file-sharing such that it can be implemented within a protocol requiring no user control [5].

2.3 Explicit and implicit utility

How is utility or fitness obtained in applications? In the abstract models utility is generally specified *a priori* with some payoff matrix of a game (such as the Prisoner's Dilemma). These models abstract away from any specific interaction

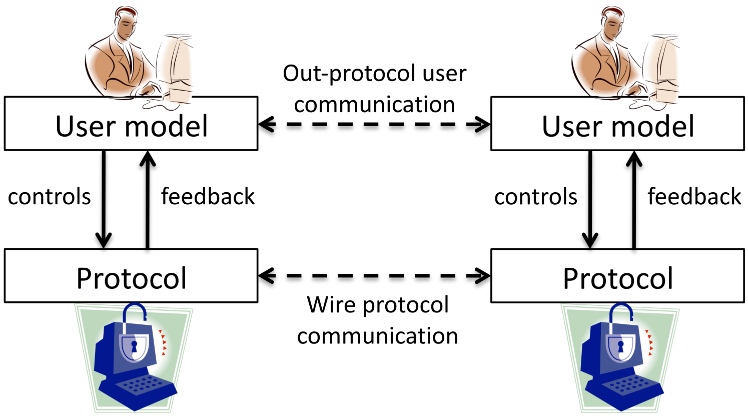


Figure 2.1: A schematic diagram showing the relationship between the user model and the protocol. Notice that the user controls the protocol (client software) and assesses its performance via feedback from the protocol software - this may be directly using a graphical user interface (GUI) or indirectly based on the the assessed performance of the client software. Minimally the user downloads and installs the protocol and can decide to terminate it or leave it running. Users can communicate out-of-protocol by using other protocols (such as e-mail) or direct communication.

producing utility (desirable outcomes for the agent) and this is what give such models potential generality and power. When we wish to produce an actual application we need to find some way of measuring utility which captures the quality of the outcome for each node. Given such a measure it is possible to assess the social utility of the system (by say, aggregating all utilities). What abstract cooperation models give us is a way to ensure that even if the agents follow some local selfish (or otherwise) utility optimising approach they can self-organise to socially beneficial outcomes (cooperate) without central control. In applications (even when well specified) it is often non-trivial to select a utility function. There are two fundamental ways to tackle this, either via some explicit function that can be built in to the protocol or some implicit function that could be determined from user behaviour. For example, in a filesharing system one might have an explicit function that gives the average download time for a node. For, say, a streaming video application one might measure the amount of time a user watches streams. Alternatively one could give the user an explicit button or rating bar to assess the quality of their service. None of this is trivial and would probably require extensive experimentation in simulation and also empirical tests to find what works.

Evolutionary game theory models assume strategies (or behaviours) spread via imitation between nodes based on utility. Depending on the user model and protocol interpretation this can be tricky in applications. If we assume users will copy the behaviours of the more successful others then we need to be clear how this happens (raising the issues of utility and plausible in-protocol and out-of-protocol methods of comparison and spread). It is possible to build protocols that perform this copying without the explicit knowledge of the user (i.e. protocol level) but this raises a number of security and utility issues. However, our planned P2P-widget infrastructure gives us the possible basis for at least experimentation with such novel approaches (i.e. fully distributed automatic spreading of protocol code variants). Details of the prototype P2P-widget infrastructure can be found in QLectives deliverable D4.3.1.

Determination and comparability of utilities raises several issues. A possible solution to the utility issue in applications is to use thresholds in the nodes (or aspiration levels) which can be dynamically adjusted so a node can determine if it is satisfied or not without requiring comparison with other nodes' utilities. This is called *Satisficing* and is a proposed model of how many humans and economic actors may actually behave [35]. Another approach relies on direct user input to determine if they are satisfied with the current quality of of service - this then becomes a user interaction issue (involving GUI and user model) - supplying the user with the necessary information and controls to produce cooperative interactions.

2.4 User attention as scarce resource

In many potential application areas an important scarce resource is the attention of the user. When a system requires user contributions (such as content or rating) it is necessary to incentivise, in some way, users to contribute this scarce resource. Much of the general cooperation literature (to our knowledge) does not address the idea of attention directly. However, there are possible interpretations incorporating utility values. For example in a public goods game, in which agents select a contribution amount, this could be interpreted as an attention contribution to the community. Another interesting link between the economics of attention and the user model / protocol distinction we made, above, is that the automation of user behaviour into a protocol is a way of requiring less user attention to achieve the same goal. For example, it is well known that media sharing communities originally formed around internet news services (and indeed still exist) but that using such methods to share large files requires a high level of user competence and attention. Hence automated systems, such as BitTorrent are more popular. We have only recently begun to think along these lines but believe this could inform interesting future work which we discuss briefly in chapter 5.

2.5 Application areas

In this section we aim to identify some broad requirements for QMedia, with the previously discussed issues in mind, that can be addressed via the user models and collective mechanisms that we will discuss in subsequent sections. Further details concerning QMedia can be found in QLectives deliverables D4.3.1.

We have not specified any QScience requirements, at this stage, but we briefly discuss them in chapter 5.

2.5.1 Promoting seeding for media sharing

In order to produce efficient media sharing and high performance download in QMedia it is necessary to provide incentives and mechanisms such that peers contribute enough upload bandwidth to supply community needs. In general upload bandwidth can be considered to be a scarce resource. This is because many users have limited upload bandwidth relative to download bandwidth and, in addition, can "spend"' this bandwidth in any way that they please. Hence it can not be assumed that all users will voluntarily give their bandwidth freely when it is required. This requirement is even more important for video-on-demand and streaming systems (which QMedia offers in addition to downloading) since a high number of upload contributors are required at all times during viewing in order to avoid pauses and stalls - which will immediately discourage users from continuing to use the system.

Currently the BitTorrent protocol (on which QMedia is based) uses a direct reciprocity approach (called tit-for-tat) to incentivise upload contributions by peers [5]. However, this only produces incentives while a peer is currently downloading a file and does not produce incentives for "seeding" a file (sharing the entire file) or for video-on-demand streaming. Additionally, the current tit-for-tat approach can be cheated via hacked clients [26, 19].

It has been recently noted that so-called BitTorrent "private communities" produce higher amounts of cooperation and particularly seeding behaviour than "open communities" [1, 14]. Private communities require explicit membership (via a central website) limiting content to members only. Also many such communities implement credit or points systems that centrally monitor and store upload and download behaviour thus incentivising seeding behaviour. However, interestingly, not all private communities implement such policies yet still show increases in cooperation. It would appear that such communities emerge socially beneficial norms via community interactions and formation mechanisms (such as invite only membership or strong affinity membership).

A requirement for QMedia is to produce the levels of cooperation (seeding) found in private communities without the need for a central website or administration. That is, we wish to provide a distributed incentive system of sufficient quality to support efficient downloading and video-on-demand.

2.5.2 Promoting quality content and metadata

QMedia aims to be a social media sharing client allowing users to find and share content with others having similar tastes and interests. Currently, BitTorrent users create communities using websites which list media items (.torrents files) being shared by other members of the community. These communities often focus around particular interests or affinities - such as media from a given country or genre. Some communities are private, meaning they require explicit membership. while others are open - meaning they can be access by anyone. In both cases the communities require centralised websites and administration. Private communities often apply membership rules and can sanction users with exclusion or penalties if those rules are broken. These communities often implement forums and messaging services allowing users to communicate and coordinate their activities - for example individuals may request particular media from the community. In addition such communities often moderate the content submitted verifying it is of high quality and attaching high quality metadata to the media - such as thumbnail pictures, ratings, comments, subtitles and descriptions. It could be argued that much of the quality BitTorrent media sharing today takes place within private communities.

We wish QMedia to provide tools to encourage quality contributions from community members. In addition to high levels of bandwidth contribution (high levels of seeding) found in many existing online media sharing communities we also observe high levels of user involvement supporting the peer production of high quality metadata, spam prevention, forum participation and content injection. Currently the tools that support this require centralised servers and administration.

A requirement for QMedia is to provide tools and incentives that promote quality user contributions in the form of rating, moderation and spam prevention without the need for a centralised website or administration. Hence we wish to produce a self-managing and self-policing system sufficient to support high levels of community quality.

2.6 Summary

In this chapter we have discussed some of the issues we encounter when moving from abstract models of cooperation, found in the literature, to application areas in peer-to-peer systems. It is clear that in performing such a translation it is necessary to carefully consider how the mechanisms and quantities expressed in the model can be interpreted within an application area. For example it is necessary to unpack agent models into user and protocol aspects. Also the interpretation of utility needs to be carefully considered. With these issues in mind we presented two high level application areas, within QMedia, that could benefit from high-levels of cooperation.

In the next chapter we present some candidate user models and collective mechanisms that may be applicable to the application areas and in chapter 4 we indicate how these can be applied.

Chapter 3

Candidate models

In this chapter we list both candidate *user models* and, what we term, *collective mechanisms*. The user models can be seen as hypotheses concerning how users will behave when they have choices - such as how much upload bandwidth to contribute or when they may join or leave a group or system. The collective mechanisms can be interpreted as templates for potential protocols that structure the interaction between users such that quality outcomes emerge. In this case we are interested in mechanisms that lead to high levels of cooperation between peers promoting the application areas discussed in the previous chapter.

3.1 User models

In this section we identify several broad user model variants that can be found in a number of theory models related to the emergence of cooperation (and other phenomena). Our aim is not to focus on one particular model but to define a range of models which we can use in simulation and potentially compare against measurements of real user behaviour. In addition to evaluating collective mechanisms with given user models we intend to consider mixtures of such models in a population - that is different proportions of different models. For example, if a collective mechanisms generates high levels of cooperation with, say, all users following an evolutionary model, we can evaluate if this still holds if a small proportion of default or rational users enter the system.

Previous work on applications of theoretical user models to P2P systems have tended to be narrow. For example, in mechanism design approaches it has been proposed that three variants should be considered: Correct / Obedient, Rational and Irrational [37]. Here we adopt the former two but unpack the irrational variant to include altruistic, evolutionary and satisificing models.

3.1.1 Default model

By default model we mean a kind of null model. It is often found that users within P2P systems appear to be doing little more than running their client software - without any significant interaction - for some period of time. Essentially nodes follow the protocol. This has been termed the Correct / Obedient model in previous work [37].

3.1.2 Rational model

A rational model indicates a user that will pursue their own self-interest using the controls they have over the protocol. Here we discount the user hacking or changing the client protocol. This can be contrasted with the definition given in [37] where local protocols can be changed.

For the purposes of experimentation we may adopt, where applicable, variants of a rational user model as a baseline case and for comparing with other models. Specifically we can evaluate our simulations with some number of rational users where this is plausible given current measurements.

3.1.3 Altruistic model

Although rarely seen in theoretical models, several measurement studies of P2P filesharing communities evidence users who appear to behave in altruistic ways. For example, seeding behaviour in public communities and the maintenance and administration of central websites for private communities. Additionally, the commenting, rating and moderation of media items in both public and private communities is often performed without any apparent incentives other than altruistic contribution to the community.

One area of experimental theory in the form of behavioural economics, where altruistic behaviour has been identified is in the area of altruistic punishment [9]. In certain experimental contexts, involving public goods games, participants are prepared to altruistically punish others who they feel are exploiting the group by not contributing fairly to the common good. Hence a form of self-policing can be supported via these kinds of user model (see section 3.2.3).

3.1.4 Evolutionary model

Evolutionary models of behaviour have been widely employed within evolutionary game theory and agent-based modelling [2, 38, 33, 15, 31, 23]. The assumption is that behaviour variants (or strategies) are copied between agents based on some measure of success (or fitness) - often termed utility. The assumption is that agents can determine how well they are performing using some metric and can also determine how well others are doing by the same metric. Agents can then copy (imitate) the strategies from those who outperform them. Additionally evolution implies some level of noise or innovation of strategies (mutation) in which agents spontaneously try a new strategy with some low probability. Strategies can involve more than just a choice in a game, such as interaction partners, location in a graph or group membership. Hence evolutionary user models can also be the basis for the emergence of social structures [8].

3.1.5 Satisficing model

Satisficing models [35] assume that agents are not attempting to maximise a utility but rather have some aspiration threshold which they will be satisfied with. This means the agent will not change their behaviour (strategy) if they consistently reach a level of performance which meets their aspiration threshold. This model requires less assumptions than both a rational model and an evolutionary model. It only requires that an agent can compare its own performance against its aspiration threshold. However, this raises the issue of how the aspiration threshold is determined and how it might adapt over time. Interestingly, some recent on-going work has addressed the concept of adaptive aspiration levels and has shown that in some circumstances this can result in similar performance to evolutionary approaches [32].

3.2 Collective mechanisms

In this section we present a number of collective mechanisms that have been proposed in theoretical models and experiments. These can be seen as potential templates for distributed protocols that, under given user models, could self-organise the system to high levels of cooperation and other quality outcomes. One way to translate the theoretical models into protocols is to identify the salient constraints (or assumptions) that support the mechanism. In this chapter we provide our initial thoughts on these. In chapter 4 we relate the assumptions to the application requirements and practical engineering constraints.

3.2.1 Indirect reciprocity

Indirect reciprocity mechanisms allow for cooperative interactions to occur between strangers who have never met before and may never meet again. This is useful in large populations with many interactions between agents that may not meet in the future. If a summary of direct reciprocity mechanisms (such as tit-for-tat) is "I'll scratch your back if you scratch mine" then indirect reciprocity can be summarised as "I'll scratch your back if you scratch his".

For example, agent A performs a beneficial act for agent B and agent C returns the favour by performing a beneficial act for agent A. Such indirect interactions can form long chains or closed loops (for example if agent B performs an altruistic act for agent C). For such systems to operate it is important that agents have access to reliable third party information concerning the previous behaviour of other agents (a reputation or image). Hence indirect reciprocity mechanisms are strongly related to reputation systems.

Within reputation systems, in general, a distinction can be made between reputation and image: Reputation involves third party information (which may or may not be correct); Image involves direct knowledge of one agent by another via some direct interaction between them or direct observation [6, 24]. The way that the two relate is determined by the methods of dissemination and aggregation of third party information employed by the agents.

In its simplest form this can involve a central trusted authority which stores the past behaviour of agents and shares this with others or supports centralised punishments and / or rewards. However, it has been shown that evolutionary models can evolve cooperation through indirect reciprocity mechanisms without central trusted third parties [23]. Such models assume some form of distributed information dissemination that transforms image into reputation often by agent-to-agent information sharing such as gossip.

The indirect reciprocity approach can be applied in systems where the following general **assumptions** hold:

- The system is composed of agents that benefit from interaction with others in the form of a public goods type game
- The population is large and unstructured such that many interactions occur between agents who have not interacted before and may never interact again
- During interactions agents have access to high quality third party information concerning the past behaviour of the other agents
- Agents can apply interaction policies such that they can punish and / or reward other agents based on the content of the third party information
- Agents desire to increase their utility or aspire to a certain performance level

The success of an indirect reciprocity mechanism is judged by how well the system self-organises toward cooperative interactions. Specifically by rewarding cooperative behaviour and / or punishing uncooperative behaviour over time through the emergence and dissemination of high quality reputation information. A good system incentivises cooperative behaviour in the agents because it is in their individual interest to obtain a good reputation so they can achieve their performance level or increase their utility.

The **emergent process** that drives indirect reciprocity models to encourage cooperative behaviours follows a, more-or-less, general pattern:

- When a pair of agents interact each forms an image of the other based on the observed outcome of the interaction (i.e. if it is cooperative or altruistic)
- Images are disseminated to other agents using some mechanism
- Images are aggregated in some way to produce a reputation
- Dissemination and aggregation processes converge producing true reputations
- It is in the interests of the agents to utilise the reputation information to apply a discriminatory policy in favour of high reputation individuals
- Hence cooperative and altruistic behaviour tends to be selected

Indirect reciprocity models may employ rational or evolutionary user models. Interestingly, altruistic user models can cause problems in some systems (see section 4.1.2).

3.2.2 Migration and group selection

Migration models allow agents to move or "migrate" (change their interaction neighbours) based on some local performance criteria [16, 17]. Migration mechanisms are highly important within the wider set of models known as group selection models [38, 15, 31, 34].

Group selection relies on the dynamic formation and dissolution of groups. Over time individual agents may change groups by moving to those that offer better individual performance. Interaction between agents, that determine performance, is mainly restricted to those sharing the same group. Essentially then, in a nutshell, groups that support high performance for the agents that comprise them grow and prosper whereas exploitative or dysfunctional groups dissolve as agents move away, or migrate, to other groups that provide better performance. Hence functional groups, in terms of satisfying individual goals, are selected over time since they are stable.

The migration / group selection approach can be applied to systems where the following general **assumptions** hold:

- The system is composed of agents that can benefit from interaction with others in the form of a public goods type game
- The population is partitioned into groups such that utility producing interaction is mainly limited to agents within the same group
- Agents determine, periodically, some performance level or utility
- Agents may spontaneously change their contribution behaviour and group membership
- Agents desire to increase their utility or aspire to a certain performance level

The success of the group selection mechanism is judged by how well the system self-organises towards achieving a collective goal (decided by the observer or designer of the system). Often this will be maximising the sum of individual performances but could involve other measures such as equality or fairness for example. In general for public goods games the aim is to maximise collective payoff - i.e. to avoid a sub-optimal equilibrium in which cooperation is low or non-existent.

The **emergent process** that drives group selection models to encourage group beneficial cooperative behaviours follows a, more-or-less, general pattern:

- Agents are grouped in some initially arbitrary way
- Interactions between agents within groups determine agent utilities
- Based on utility comparisons between agents, and possibly randomized change, group memberships and interaction behavior (strategy) change over time
- Groups which produce high utility for their members tend to grow and persist as agents join
- Groups which produce low utility for their members tend to disperse as agents leave

- Hence group beneficial behavior tends to be selected

Often group selection and migration models employ an evolutionary user model combined with some migration rule. However both satisificing [32] and rational [40] user models can be applied under certain conditions.

3.2.3 Altruistic punishment

Experimental evidence [9, 10] shows that adding the possibility of altruistically punishing defectors to public-goods games increases significantly cooperation levels. An altruistic punishment happens when an agent decides to incur some cost to reduce the benefit a non-cooperator agent receives in the game.

Threatening uncooperative agents with punishment has a somewhat intuitive effect in the game: if the punishment mechanism is efficient enough, cooperating becomes the dominant strategy. Indeed, the positive effect that institutions for monitoring and sanctioning rule-breaking have on cooperation are observed in a variety of settings [25]. The counter-intuitive result in the experiments with altruistic punishment is that agents punish non-cooperators in spite of the cost this represents to punishers.

Because there is a cost for punishing, altruistic punishment creates a second-order public-goods game: each subject can opt to cooperate by punishing the uncooperative agents; all subjects benefit from the punishers' effort. The observation that this second-order good is efficiently provisioned in laboratory conditions creates a puzzle: why would subjects cooperate in this game more than in the first-order game?

From a behavioral perspective, one possible explanation is that human beings are culturally or biologically prone to punish. Fehr and collaborators have documented that altruistic behaviour is common on the provision of punishment, potentially because of the emotional effects of free riding and punishment [9].

From an evolutionary perspective, Boyd et al. [4] show that cooperation prevails for a number of conditions if altruistic punishment is combined with group selection. This happens because groups with more punishers produce higher levels of cooperation, which, in turn, reduce the cost that punishers incur. If punishment is common, this mechanism can lead to sustained high levels of punishment. It is worth noting that, at the same time, group selection can promote cooperation for a wider range of parameters if combined with altruistic punishment.

The altruistic punishment approach can thus be applied to systems where the following **assumptions** hold:

- Agents have the capacity to detect defection and to punish defectors, although detection may not be perfectly accurate.
- Punishment cannot be done in a centralized or automated manner; it needs agents' abilities to detect or punish defection.

- If cooperation is to emerge through an evolutionary process, the assumptions for a group selection mechanism hold.
- If cooperation is to emerge in a public-goods game, some agents have the propensity to punish as an innate trait.

The **emergent process** that drives altruistic punishment models to promote cooperation follows one of two patterns. If considering a public-goods game played by human agents and no evolution:

- After the outcome of a round in the game is known by the agents, agents can identify whether there were non-cooperators;
- each agent chooses whether he/she wants to invest some of its budget to punish the non-cooperators;
- upon being punished, some of these non-cooperators change their strategy to cooperation;
- there is a fraction of the agents who reciprocate both defection and cooperation, and start to cooperate once levels of cooperation are sufficiently high;
- cooperation therefore prevails.

If considering an evolutionary setting were group selection and altruistic punishment co-exist, the process is similar to the one that happens in group selection, adding some more steps:

- Agents are grouped in some initial way: the size of groups may be considerably larger than in group selection mechanisms.
- Agents interact with their group periodically.
- After each interaction, punishers punish defectors.
- Agents meet randomly agents from other groups and depending on their relative payoff in the game, decide to imitate their strategy.
- Groups periodically engage in disputes that result in group replacement; the probability of winning a dispute is proportional to the amount of cooperation in the group.
- Altruistic punishment becomes sustainable in groups where punishers are common enough and defection does not happen very often. In these groups, punishers have little extra cost compared to cooperators.
- As group selection works in favour of groups with punishers, which are more resistant to defectors, these groups thrive.

On an evolutionary perspective, altruistic punishment models employ and evolutionary user model combined with a migration rule. As in group selection mechanisms, rational and satisficing user models could also be assumed, although with non-obvious implications. When considering a public-goods game without evolution, user models must include some degree of altruism.

Chapter 4

Application of models

4.1 Indirect reciprocity for seeding

In section 2.5.1 we presented a QMedia application domain that requires high levels of cooperative seeding (that is sharing of bandwidth) by peers. We wish to produce the high levels of seeding often found in private communities to support fast downloading and video-on-demand. In section 3.2.1 we listed a set of assumptions and the emergent process that characterises indirect reciprocity mechanisms in general. Below we relate the domain to the mechanism assumptions and then present some on-going work in this area.

4.1.1 Assumptions

The system is composed of agents that benefit from interaction with others in the form of a public goods type game.

Seeding behaviour in repeated swarm interactions: In BitTorrent peers interact in swarms to share files. BitTorrent provides no incentive to seed a file. Yet seeding improves performance for all in the swarm. Hence the amount of seeding a peer performs can be interpreted as a contribution to the public good. However a single swarm is not strictly comparable to a public goods game because the decision to seed can not affect the payoff of the seeder - since they have completed the "game" already - they have the entire file by definition. However, over some time period, assuming repeated interaction over different swarms, then this can be interpreted as similar to a public goods game. The amount of seeding each peer provides, on average, becomes a contribution amount to the community. There is a multiplier effect to the seeding contribution in a swarm because those downloading from the seeder share parts of the file with others who are downloading.

The population is large and unstructured such that many interactions occur between agents who have not interacted before and may never interact again.

Low rendezvous probability: Although peers interact repeatedly in different swarms it is generally highly unlikely they will meet the same peers again from previous swarms. If this was the case then direct reciprocity methods could be used (as BitTorrent does for the sharing of pieces of the file within a swarm).

During interactions agents have access to high quality third party information concerning the past behaviour of the other agents.

Private tracker and BarterCast: We present on-going experiments that assume a central store of reputation (as found in Private Trackers in private communities) and a distributed store of reputation (a system called BarterCast implemented in Tribler). Reputation is based on past seeding behaviour. The higher the seeding the better the reputation. Both of these approaches aim to supply high quality third party reputation information.

Agents can apply interaction policies such that they can punish and / or reward other agents based on the content of the third party information.

Refuse download to a peer: Peers with poor reputations based on past seeding contribution can be refused upload from others or can be denied entry into swarms by private trackers.

Agents desire to increase their utility or aspire to a certain performance level.

Users desire fast downloads: We assume that users desire a certain quality of service in terms of speed to download a file. For streaming of files this requires a continuous minimum performance otherwise the stream will stall.

4.1.2 Credit dynamics in centralised private communities

We have produced some initial work which assess the effectiveness of simple credit policies over some simple user model variants including satisficing, default and altruistic. We found that intuitive credit policies often lead to inefficient outcomes given these user model variants. These approaches assume a centralised trusted accounting and punishment system yet even future fully distributed credit systems would suffer from similar problems (compare the recent work in [21]). Detailed results can be found in the related papers [14, 30] but here we give a brief summary.

Crunches and crashes

Many private peer-to-peer file sharing communities implement credit policies to incentivise users to contribute upload resources. Such policies implicitly assume a user model - how the user controlling each peer behaves. We showed using

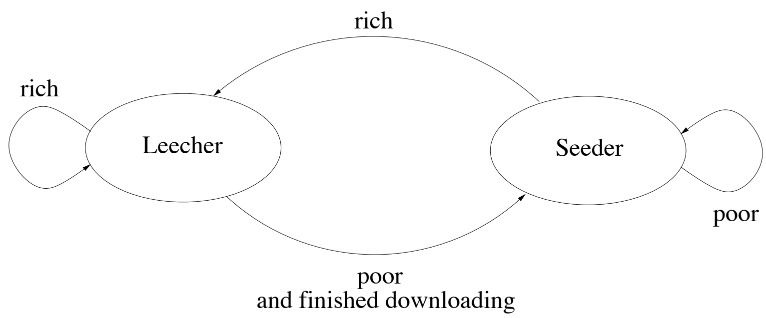


Figure 4.1: A state transition diagram indicating how peers move between seeding and leeching sessions in the credit dynamics simulation. This is a satisficing user model in which "rich" = satisfied and "poor" = not satisfied. In this case the aspiration level was exogenous and fixed.

an agent-based simulation that credit policies, based on bandwidth contribution, and *satisficing, altruistic and default* user models, can lead to both "crunches" and "crashes" where the system seizes completely due to too little credit or too much credit. We explored the conditions that lead to these system pathologies and presented a theoretical analysis that allowed us to determine if a community is sustainable or will eventually crunch or crash. We also applied the analysis to produce a novel adaptive credit system that automatically adjusts credit policies to maintain system sustainability (under the assumption of a satisficing model).

Figure 4.1 shows a diagram representing the satisificing user model. In this case we assumed that a user would stop contributing bandwidth when it considered itself to be "rich" meaning that it had enough credit to download another media file. Table 4.1 shows some simulation results where all users follow this satisficing model and where different proportions of the initial population are given enough credit to be in the rich state (i.e. satisfied - and hence not seeding to other peers). Without going into full detail of the meaning of the table we can identify a number of interesting results. Firstly, if there is too little or too much credit in the system (i.e. if the proportion of rich at start is ≤ 0.2 or ≥ 0.8) then the final state of the system is a crunch (where all peers are seeding) or a crash (where no peers are seeding). This leads system throughput (amount of data exchanged between peers and hence a measure of system performance) to go to zero. Too much seeding is just as bad as too much leeching (purely downloading) because for seeding to increase system throughput there must be a matching peer who wishes to download. Notice that high amounts of seeding (in the case of 0.5 proportion of rich at start) produces less throughput than the lower levels of seeding (when initial proportion of rich = 0.7) in some cases.

Interestingly, we also found (results not shown here) that even when a small number of users followed an altruistic user model (in which users contribute far more bandwidth than necessary to support their downloading requirements) then this degrades the performance of the system as a whole due to the hogging

Table 4.1: Results for satisficing peers with constant total credit. The main observation here is that system is only sustainable when there is not too little or too much initial credit (prop. of rich at start). Notice also that throughput (system performance) varies also buy initial credit.

prop.of **rich** at start	avg. **throughput** (std.dev)	avg. prop. of **seeders** (std.dev)	final **state**
0.1	0.0003 (0.0000)	1.0000 (0.0000)	crunch
0.3	0.2183 (0.0014)	0.9525 (0.0014)	sustain
0.5	0.7769 (0.0023)	0.7685 (0.0023)	sustain
0.7	0.9684 (0.0036)	0.5064 (0.0036)	sustain
0.8	0.5867 (0.4780)	0.2485 (0.4780)	sustain/crash
0.9	0.0008 (0.0000)	0.0000 (0.0000)	crash

of credit. This can be compared to the idea of credit hoarding as identified in [18]. This indicates that simple credit systems function poorly even if a small number of users behave altruistically - which is counterintuitive and less than desirable.

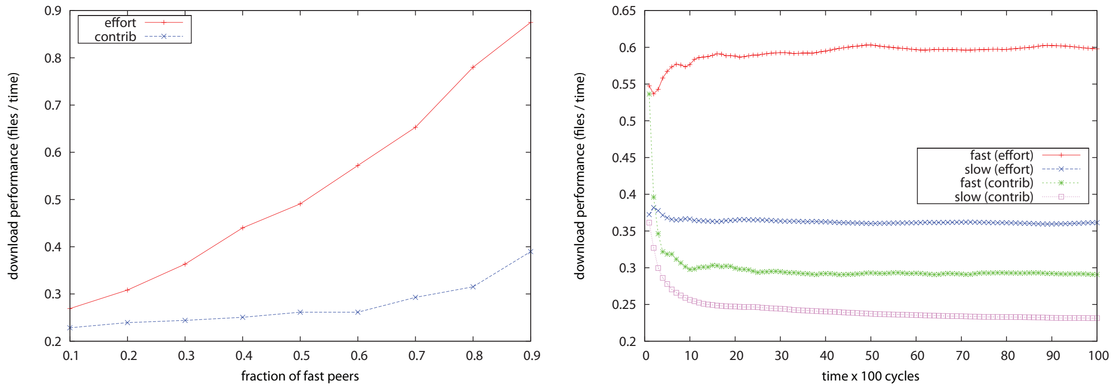
Effort-based policies

Some on-going work has considered alternative credit policies, based on effort rather than contribution, which can ameliorate some of the problems associated with inequality within private communities [29, 27]. Essentially such an approach rewards peers with credit based on the amount of effort (what proportion of available resources are contributed) rather than total contribution (what amount of resources are contributed). This means the user is rewarded for good behaviour rather than just for absolute contribution. Some simulation results evaluating an effort based approach under the assumption of a default user model (with both fast and slow upload bandwidth peers) are shown in figure 4.2. Notice that under the effort based based policy both fast and slow peers perform better than in the contribution based approach. Full details of these simulations can be found in [27].

One limitation to deployment of effort based approaches is that it is difficult in many circumstances to design methods for determining accurately what the effort value is since this requires knowledge of the actual endowments of each peer although recent work may offer a potential way to approach the problem [36].

4.1.3 Reputation in decentralised communities

Since we are interested in producing fully decentralised systems we have been experimenting with an already deployed distributed reputation system within Tribler called BarterCast [22]. Although deployed and yielding measurements



(a) (b)

Figure 4.2: Various results of contrib and effort Credit Based Scheme simulations (a) avg. download performance of all peers; (b) the download performance over time of both slow and fast peers, in equal proportion. These results are taken from [27].

BarterCast does not currently affect client performance because no policies are implemented over it. We aim to assess, for given user model variants, a number reputation based policies possibly including the application of betweenness centrality measures (see below). Firstly we give a brief overview of BarterCast, full details are given in [22].

BarterCast

BarterCast is a fully decentralised algorithm in which nodes store past upload and download amounts made to other nodes while sharing files. Periodically nodes are paired randomly using the gossip based overlay protocol built into Tribler. When nodes are paired they exchange their local store of upload and download amounts to other nodes (hence are supplying third party information). As a security measure nodes do not pass data obtained from other nodes but only their own image information on other nodes obtained by their direct experience.

Hence, in the terminology expressed in section 3.2.1, nodes exchange image data but not reputation data. This limits the spread of incorrect data, such as nodes lying to increase their reputation, because only direct reports - image - from other nodes is accepted. This does not mean that a node can not lie concerning its direct interactions: reporting, say, that it has uploaded more or downloaded more from some other node. However, BarterCast limits the effect of such incorrect information by using a maxflow approach which was previously suggested in [12]. Essentially, each node builds a reputation graph based on its own direct experiences (image) and the received images from other nodes (reputation).

The BarterCast graph contains nodes (representing all peers seen) linked by weighted edges representing the upload and download amounts between them.

A node can calculate the reputation of another node by finding the maximum flow between itself and the other node in both directions - that is, finding a route between the nodes with the maximum value that could flow over the weighted edges. This allows for a calculation to be made as to the “goodness” of the node in terms of how much upload relative to download it has given to the system based on the previous reports received and its own direct experience. It is not easy to cheat such a system because all flows must go through, initially, the direct experience (or image) of the node which is calculating the flow. However, this means that calculated reputations are subjective and hence can vary between nodes.

When a node encounters another that requests upload, during file sharing, it can calculate the reputation of the node and decide if it will grant the requested upload or reject it. In this way nodes with low reputation can be punished. However, this is not currently implemented in Tribler.

BarterCast II

One issue which has been found by measurement of the deployed system is that many nodes do not have connections to any other nodes in the graph - see figure 4.3. Also the average subjective reputations calculated by nodes is far from ideal - see figure 4.4. This could be due to peers not staying in the system for long enough (and not being able to build up a reputation graph) and / or not sharing files with other Tribler clients.

One way that may be employed to tackle this, which will be incorporated into the emerging BarterCast II protocol, is to relax the constraint that nodes always incorporate their own image information when calculating a reputation. By selecting a highly connected node as a “proxy”’ from which to start the maxflow process, nodes with no image information (direct experience) can form a maxflow to, and hence reputation for, most other nodes. This means treating the reported images from the proxy node as personal images (i.e. direct experience).

There are many possible ways a node could choose a proxy from its BarterCast graph. One possible method under investigation is to calculate the betweenness centrality value of all nodes in the graph and chose as proxy the highest valued node. Betweenness centrality measures, for a given node, the number of shortest paths between all other pairs of nodes, in the graph, that pass through the given node. So for example, in a star topology with a central node connecting all other nodes the Betweenness centrality of the central hub node would be maximum since all other pairs of nodes have a shortest path which passes through the hub.

The idea here is that nodes with a high betweenness centrality value represent highly connected nodes with many links to other nodes which means that they are of value as proxies since they can find paths to many nodes and hence produce reputation values. Secondly, from a security perspective such nodes are likely to have contributed a lot already to the system, and this has been reported as such by third parties, and hence may be viewed as more trustable than some arbitrary node.

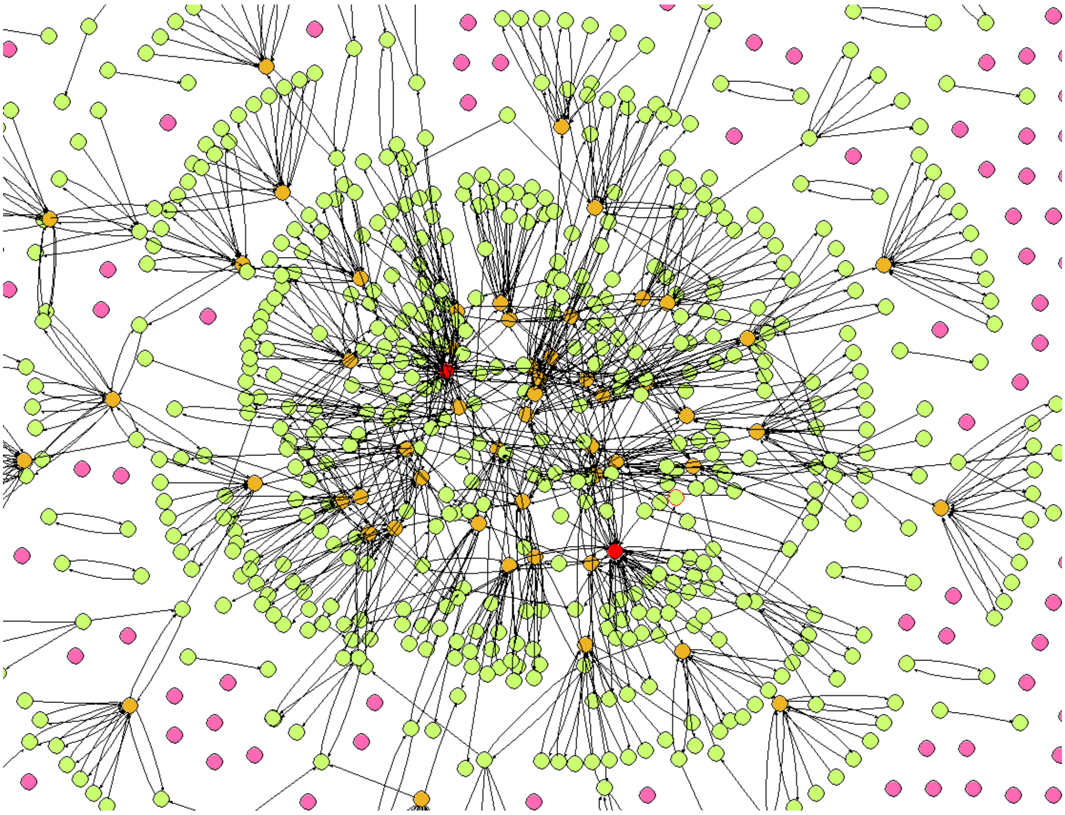


Figure 4.3: The BarterCast graph extracted from the measurement of the deployed Tribler P2P system. Notice that many nodes are not connected (in pink) and some nodes (in red) are highly connected. The large connected component evidences a scale-free structure in which experienced nodes are highly connected. The evolution of such a structure can be modelled via a preferential attachment algorithm [3]. Graph taken from [7].

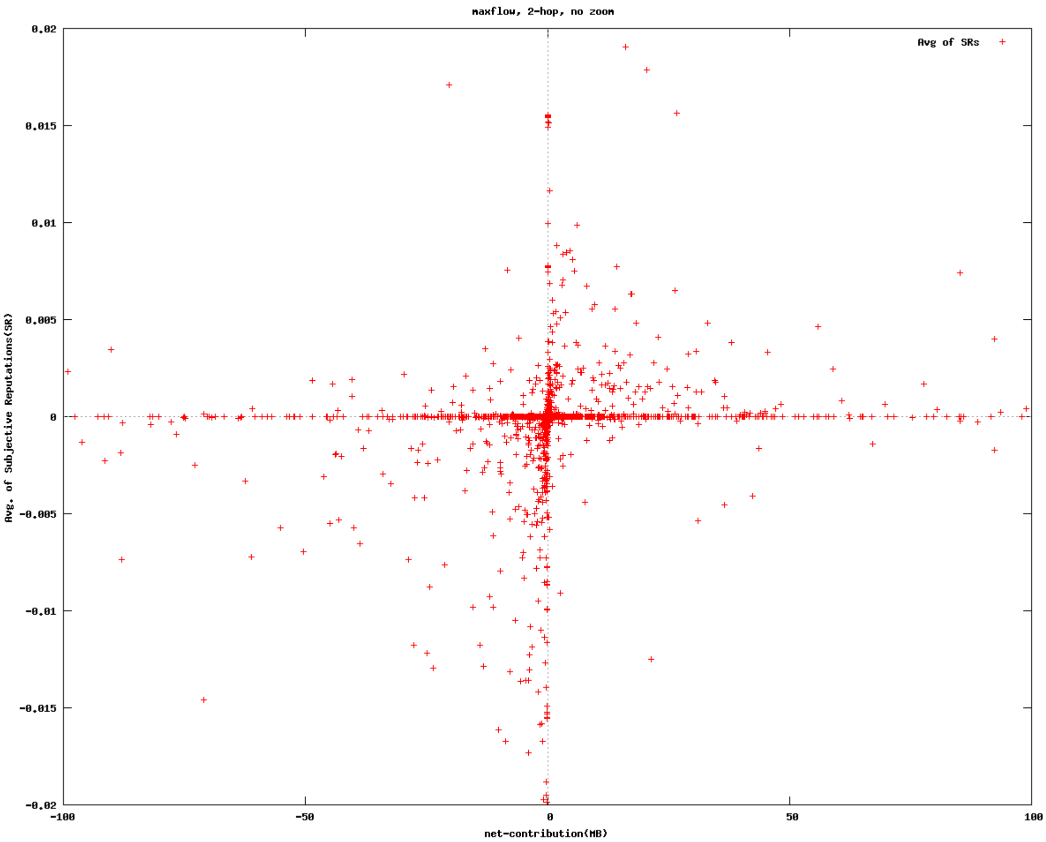


Figure 4.4: Plot showing the correlation between actual upload amount (on the x-axis) and average subjective peer reputation (on the y-axis) using the BarterCast protocol. Notice that the top-left and bottom-right quadrants are less populated than their converse quadrants. This indicates that BartCast works. However, it is evident that the system produces a high degree of noise. A perfect reputation system would produce a line of points at a 45 degree angle through the origin from bottom-left to top-right. Plot taken from [7].

It is on-going work to formulate an efficient incremental algorithm for computing betweenness centrality and also to test this approach initially with simulation using both traces from the existing measurements and possibly some common attack scenarios in which nodes seek to manipulate their reputations by giving false information.

4.2 Migration / group selection for quality content and metadata

In section 2.5.2 we presented a QMedia application domain that requires high levels of quality community contributions by users. Specifically we wish to provide tools that support the submission of high quality metadata. In section 3.2.2 we listed a set of assumptions and the emergent process that characterises migration and group selection mechanisms in general. Below we relate the quality community domain to the assumptions of the mechanism and then present some on-going work in this area.

4.2.1 Assumptions

The system is composed of agents that can benefit from interaction with others in the form of a public goods type game.

Metadata contributions by users: A quality media sharing community requires users to contribute metadata in the form of links to new content (content injection) and moderations or evaluations of this content. Making such contributions requires an effort by the user and is costly. Also users may post spam or noise contributions which reduce the quality of the community. Over some time period this can be viewed as a public goods game. High quality contributions improve experience for the whole community but there is often no incentive to do so since all benefit from good contributions.

The population is partitioned into groups such that utility producing interaction is mainly limited to agents within the same group.

Every user can create and subscribe to channels using ChannelCast:. ChannelCast is a new protocol that will be deployed in Tribler in the next release. Any user can post (inject) content into a personal channel that others can view, moderate and comment on. Users are limited to being members of only a small number of channels at any one time.

Agents determine, periodically, some performance level or utility.

Quality content and metadata: Users determine their utility or performance by the quality of the content and metadata they derive from the channels they subscribe to.

Agents may spontaneously change their contribution behaviour and group membership.

Move between channels and vary contribution: Users have the ability to leave and join channels at any time and determine their own metadata contribution to each channel they are a member of.

Agents desire to increase their utility or aspire to a certain performance level.

Users desire high quality content and metadata: We assume that users desire a certain quality of service from channels in terms of quality of content and the metadata associated with it.

4.2.2 Formation of quality channels

Tribler will soon be augmented with a new protocol called ChannelCast (produced within the P2P-Next EU project). This allows any user to create a single "channel" to which they can inject pointers to media content (.torrent files). Any user can view available channels and subscribe to some number of channels (currently set to a maximum of 8). Subscribers may view the content posted to the channel and also may attach metadata to the content. In addition metadata in the form of comments can be attached to the channel (using a protocol under development called CommentCast - see QLectives deliverable D4.1.1). These tools give the users the potential to create a dynamic ecology of groups and migration possibilities between groups. It is an open issue, at this stage, if these tools will be sufficient to support a migration / group selection process which promotes quality channels.

In order to determine the effectiveness of this approach we have two on-going lines of work:

Deploy ChannelCast and CommentCast and measure the user behaviour with respect to their use. The creation of a channel containing high quality content requires some level of altruism from the channel creator (or owner). It is an open issue if there is sufficient altruism in the system to create a sizeable population of channels. Recent centralised approaches allowing users to create similar groups appear to be successful [42] hence it will be of interest to measure how many quality channels are created over time. Another question is whether significant migration processes occur. Users need to periodically migrate between channels based on perceived quality. Finally it will be necessary to determine if sufficient contributions to the quality of channels are provided by subscribers in the form of submission of quality metadata.

Formulate a migration / group selection simulation model that captures the channel dynamics.. The channel formulation of groups deviates in some important respects from previous migration / group selection models. Firstly, a single user is responsible for a large part of the quality of the channel, they create the channel and furthermore they are solely responsible for submitting content to the channel. Hence they have a disproportionate effect on channel quality and would appear to have little incentive to do this for any user model other than altruistic. Also users may join more than one channel at a time rather than be limited to a single group. We aim to formulate a model that captures these unique features as simply as possible and then assess the conditions on which cooperative group selection would occur. For example, what proportion of the population would need to behave altruistically in a population of satisficers in order to sustain quality quality channels? How would the maximum number of channels that can be joined by a single user effect cooperation levels? What migration rules would be sufficient to ensure that a small number of spamming users could not degrade the entire system? We would hope that the model can inform and be informed by the deployed protocols.

4.3 Altruistic punishment for quality content and metadata

In chapter 2 we presented two QMedia application domains that require community contributions by users (seeding and metadata). In section 3.2.3 we listed a set of assumptions and the emergent process that characterise altruistic punishment mechanisms in general. Below we relate the seeding and quality community domains to the assumptions of the mechanism and then present some potential future work in this area.

4.3.1 Assumptions

Agents have the capacity to detect defection and to punish defectors, although detection may not be perfectly accurate.

Community contributions by users, in the form of metadata and seeding, can be viewed by other users in some way. This could involve tagging, on screen, all metadata contributions with a user identifier. Also a page could display, for a given community (a channel say), contributions made to that community by users. This could include various statistics such as date joined, comments posted, total bandwidth contribution.

Punishment cannot be done in a centralized or automated manner; it needs agents' abilities to detect or punish defection.

Users have the ability to directly punish selected others. Since quality metadata involves a semantic assessment there is no simple automated approach to promoting it. This requires direct evaluation by users. Punishment could be achieved by giving users the ability to select another user via clicking on their identifier and selecting a "slap down" option. In its simplest form this could involve the punisher refusing to upload to, or pass on metadata from, the punished for some period of time (say 24 hours). In addition this temporary blacklisting could be gossiped to other members of the group who may also punish. This might involve some form of aggregation of "slap downs" - say, three strikes and you're out. In such a situation the cost to the punishing user is the attention they need to devote to selecting users for punishment. However, a malicious user may simply select random others to punish. It would hence be of value to allow users to also see who is punishing whom and be able to punish those who appear to be punishing wrongly (meta-punishment). In addition, an explicit cost could be applied to the punisher to avoid random punishment behaviour (although it is not clear at this stage how this could be easily achieved).

If cooperation is to emerge through an evolutionary process, the assumptions for a group selection mechanism hold.

Altruistic punishment may be combined with migration / group selection of communities. If a migration / group selection process is active (see section 4.2) then it may not be necessary to assume some proportion of altruistic users.

If cooperation is to emerge in a public-goods game, some agents have the propensity to punish as an innate trait.

Some proportion of users are assumed to be altruistic. We assume that contribution to the community can be viewed as a public goods game, since members of the community may benefit from quality contributions without contributing themselves. We assume there will be enough users predisposed to altruistically punish in the population to support the emergent process.

4.3.2 Community exclusion

Punishment can be effected in a community by excluding users from certain community resources or from the community as-a-whole. The former would appear easier to implement in a fully distributed way than the latter. However both could be considered. Currently we do not have protocol designs or potential simulation models to implement such punishments. However, possible future work could consider two possibilities:

Allow individual users to punish other users directly. On way to achieve this, would be to allow for any individual user to select another for punishment (a "slap down"). This would involve temporarily blacklisting the user such that community related activities - such as posting of metadata or obtaining bandwidth for download - would be refused by the punishing user over some period of time. However, this approach would require a significant proportion of members of the community to decide to punish to have a significant effect on the punished individual.

Allow the aggregation of community punishment requests allowing for collective action to exclude an individual from the community. This would involve some method for aggregation of punishment requests within a community. One possibility is for community members to communicate punishment evaluations to other members over time and when some threshold is reached then punishment is enabled for some period. This is similar to an indirect reciprocity scheme in which punishment decisions are aggregated rather than image. A Tribler protocol that could be adapted for this purpose is the already deployed VoteCast [28] protocol. In general we may explore bottom-up mechanisms of "punishment institutionalisation".

Chapter 5

Summary and Further Research Questions

In this deliverable we have presented some initial high-level QMedia application domains that would benefit from high levels of cooperation. We also selected a set of user models and collective mechanisms inspired by a number of abstract cooperation models. We related those models to the application domains through specifying how the assumptions of the models can be interpreted within the application domains.

We have outlined recent and on-going work considering credit dynamics in private communities with centralised accounting and enforcement mechanisms and fully distributed reputation systems.

We have shown that credit systems can lead to highly sub-optimal outcomes for given user models in simulated file-sharing environments and have suggested some possible approaches to addressing these involving both dynamic credit policies and non-conventional (effort based) incentive schemes.

On-going work involves the measurement of BarterCast and the simulation of the BarterCast II protocol involving the development of an efficient incremental betweenness centrality algorithm. Additionally simulation may be used to test different incentive policies within BarterCast II over various user models.

We identified two lines of on-going work applying the migration / group selection mechanism. These involve deploying and modelling the effect of the ChannelCast and CommentCast protocols on collective user behaviour. These protocols provide a minimal way of users creating, migrating and contributing to groups. Some of the application constraints differ from the abstract models (e.g. all groups are created and owned by a single user and users may join several groups at one time) and this could be tested within simulation.

We presented two possible future lines of work utilising an altruistic punishment mechanism. In each case we envisage community exclusion approaches that allow users to punish others through limiting their access to community resources. Firstly, users may apply individual punishments to other users directly and secondly some aggregation mechanisms could be used that allows third party information to inform punishment. Both these approaches would require new protocols and these could benefit from simulation models.

We also believe that linking to work on the “economics of attention” may be a productive way to refine simulation models applicable to application areas requiring significant user attention to function correctly [41]. Additionally such models of attention would appear to be applicable to quality scientific collaboration and community formation [20]. We are at an early stage in considering this approach but this could lead to more refined user models incorporating dimensions of attention.

We have not, at this stage, applied analysis to the QScience application domain. However, it appears plausible that the promotion of quality communities within QMedia has some similarities to that required by a scientific community, e.g. the need to provide quality metadata (such as ratings, reviews and comments). Future work could address this in more detail.

Finally, our aim in this deliverable has been to select some initial promising candidate models, state their key assumptions and relate them to possible deployable application domains. These models and domains are not exclusive and future work may examine others where applicable.

Bibliography

[1] Andrade, N. and Santos-Neto, E. and Brasileiro, F. and Ripeanu, M. (2008) Resource demand and supply in BitTorrent content-sharing communities. *Computer Networks*, 53:515–527.

[2] Axelrod, R. (1984) *The Evolution of Cooperation*, Basic Books.

[3] Barabási, A.-L., Albert, R., (1999) Science 286, 509-512.

[4] Boyd R, Gintis H, Bowles S, Richerson P. J. (2003) The evolution of altruistic punishment. Proc Nat. Acad Sci USA 100(6):3531-3535.

[5] Cohen, B. (2003) Incentives build robustness in BitTorrent. In *Workshop on Economics of Peer-to-Peer Systems*.

[6] Conte R., Paolucci M. (2002). Reputation in Artificial Societies. Social Beliefs for Social Order. Boston: Kluwer.

[7] Delaviz, Rahim - BarterCast Measurements. See: https://www.tribler.org/trac/wiki/SwarmBasedReputationConsensus

[8] Edmonds, B., Norling, E., Hales, D. (2009) Towards the Evolution of Social Structure. Computational and Mathematical Organization Theory (CMOT), Vol. 15 No. 2. June 2009. doi:10.1007/s10588-008-9052-2

[9] Fehr, Ernst, Fischbacher, Urs and GŁchter, Simon (2002). Strong Reciprocity, Human Cooperation and the Enforcement of Social Norms. Human Nature 13, 1-25.

[10] Fehr, Ernst and Gintis, Herbert (2007). Human Motivation and Social Cooperation: Experimental and Analytical Foundations. Annual Review of Sociology, Vol. 33.

[11] Fehr, Ernst and GŁchter, Simon (2002). Altruistic Punishment in Humans. Nature 415, 137-140.

[12] Feldman, M. et al., (2004) Robust Incentive Techniques for Peer-to-Peer Networks, Proc. 5th ACM Conf. Electronic Commerce (EC 04), ACM Press, 2004,pp. 102111.

[13] Garbacki, P. and Epema, D.H.J. and van Steen, M. (2007) An Amortized Tit-For-Tat Protocol for Exchanging Bandwidth instead of Content in P2P Networks. In *Proc. First International Conference on Self-Adaptive and Self-Organizing Systems*, pp.119–128.

[14] Hales, D., Rahman, R., Zhang, B., Meulpolder, M., Pouwelse, J. (2009) BitTorrent or BitCrunch: Evidence of a credit squeeze in BitTorrent? In *Proc. IEEE International Workshops on Enabling Technologies: Infrastructures for Collaborative Enterprises*, pp.99–104.

[15] Hales, D. and Arteconi, S. (2006) SLACER: A Self-Organizing Protocol for Coordination in P2P Networks. IEEE Intelligent Systems, 21(2):29-35.

[16] Helbing, D. (2009) Pattern formation, social forces, and diffusion instability in games with success-driven motion, Eur. Phys. J. B 67, 345 (2009).

[17] Helbing D. and Yu, W. (2009) The outbreak of cooperation among success-driven individuals under noisy conditions, PNAS 106, pp. 36803685

[18] Kash, I.A. and Friedman, E.J. and Halpern, J.Y. (2007) Optimizing scrip systems: Efficiency, crashes, hoarders, and altruists. In *Proc. ACM Conference on Electronic Commerce*, pp.305–315.

[19] Locher, T. and Moor, P. and Schmid, S. and Wattenhofer, R. (2006) Free Riding in BitTorrent is Cheap. In *Proc. 5th Workshop on Hot Topics in Networks (HotNets)*.

[20] Lada A. Adamic, Xiao Wei, Jiang Yang, Sean Gerrish, Kevin K. Nam, Gavin S. Clarkson (2010) Individual focus and knowledge contribution. arXiv:1002.0561v1

[21] Landa, R. and Griffin, D. and Clegg, R. and Mykoniati, E. and Rio, M. (2009) A sybilproof indirect reciprocity mechanism for peer-to-peer networks. In: 2009 Proceedings IEEE INFOCOM 19 25 April 2009 Rio de Janeiro, Brazil. The Institute of Electrical and Electronics Engineers, Piscataway, US, pp. 343-351

[22] Meulpolder, M., Pouwelse, J.,, Epema, D., Sips, H. (2009) BarterCast: A practical approach to prevent lazy freeriding in P2P networks, ipdps, pp.1-8, 2009 IEEE International Symposium on Parallel and Distributed Processing.

[23] Nowak, MA; Sigmund, K. (2005) Evolution of indirect reciprocity. Nature, 437:12911298.

[24] Nowak, MA; Sigmund, K. (1998) Evolution of indirect reciprocity by image scoring. Nature, 393:573-577.

[25] Ostrom, E.; Gardner, R. and Walker, J. Rules, Games, and Common-Pool Resources University of Michigan Press, 1993

[26] Piatek, M. and Isdal, T. and Anderson, T. and Krishnamurthy, A. and Venkataramani, A. (2007) Do incentives build robustness in BitTorrent? In *Proc. of NSDI*.

[27] Rahman, R., Meulpolder, M, Hales, D., Pouwelse, J., and Sips, H. (2010) Improving Efficiency and Fairness in P2P Systems with Effort-Based Incentives. To be published in Proceedings of the IEEE International Conference on Communications, 23-27th May 2010, Cape Town, South Africa.

[28] Rahman, R., and Hales, D., Meulpolder, M., Heinink, V., Pouwelse, J., Sips, H. (2009) Robust vote sampling in a P2P media distribution system. Proceedings IPDPS 2009 (HotP2P 2009), IEEE Computer Society.

[29] Rahman, R., Meulpolder, M, Hales, D., Pouwelse, J., and Sips, H. (2009) Revisiting Social Welfare in P2P. Technical report PDS-2009-003. Parallel and Distributed Systems, Technical University of Delft, 2009.

[30] Rahman, R., Hales, D., Vinko, T., Pouwelse, J., Sips, H. (forthcoming) No more crash or crunch: sustainable credit dynamics in a P2P community.

[31] Riolo, R., Cohen, and M.D., Axelrod, R. (2001) Cooperation without Reciprocity. *Nature* 414, 441-443.

[32] Roca, C.P. and Helbing, D. (personal communication).

[33] Rossi, G., Arteconi, S., Hales, D. (2009) Evolving Networks for Social Optima in the Weakest Link Game. Computational and Mathematical Organization Theory (CMOT), Vol. 15 No. 2. June 2009. doi:10.1007/s10588-008-9051-1

[34] Santos F. C., Pacheco J. M., Lenaerts T. (2006) Cooperation prevails when individuals adjust their social ties. *PLoS Comput Biol 2(10): e140. DOI: 10.1371/journal.pcbi.0020140*

[35] Simon, H. (1955) A Behavioral Model of Rational Choice. The Quarterly Journal of Economics, Vol. 69, No. 1. (Feb., 1955), pp. 99-118

[36] Snader, R. and Borisov, N. (2009) EigenSpeed: Secure Peer-to-peer Bandwidth Evaluation. In *8th International Workshop on Peer-to-Peer Systems (IPTPS'09)*.

[37] Shneidman, J. and Parkes, D.C.(2003) Rationality and Self-Interest in Peer to Peer Networks. In Peer-to-Peer Systems II. LNCS 2735, Springer.

[38] Traulsen, A. and Nowak, M.A. (2006) Evolution of cooperation by multilevel selection, PNAS 103, pp. 10952–10955.

[39] Trivers RL (1971) Evolution of reciprocal altruism. Q Rev Biol 46(1):35-57.

[40] Tiebout, C. (1956) A Pure Theory of Local Expenditures. The Journal of Political Economy, 64(5):416-24.

[41] Wu, F. and Huberman, B., A. (2007) Novelty and collective attention PNAS vol. 104 no. 45 17599-17601

[42] The hexagon system: http://hexagon.cc/