INCENTIVE-DRIVEN QOS IN PEER-TO-PEER OVERLAYS

ENGINEERING RESOURCE ALLOCATION TECHNIQUES FOR LIVE, PEER-TO-PEER STREAMING

Raúl Leonardo Landa Gamiochipi

UCL

Department of Electronic and Electrical Engineering

Communications and Information Systems Group

Supervisor: Dr. Miguel Rio

A dissertation submitted in partial fulfilment of the requirements for the degree of Doctor of Philosophy of University College London

UCL

· May 8, 2010·

Declaration of Originality

I, Raúl Leonardo Landa Gamiochipi, confirm that the work presented in this thesis is my own. Where information has been derived from other sources, I confirm that this has been indicated in the thesis.

(Raúl Leonardo Landa Gamiochipi, PhD Candidate)

Abstract

A well known problem in peer-to-peer overlays is that no single entity has control over the software, hardware and configuration of peers. Thus, each peer can selfishly adapt its behaviour to maximise its benefit from the overlay. This thesis is concerned with the modelling and design of *incentive mechanisms* for QoS-overlays: resource allocation protocols that provide strategic peers with participation incentives, while at the same time optimising the performance of the peer-to-peer distribution overlay.

The contributions of this thesis are as follows. First, we present *PledgeRoute*, a novel contribution accounting system that can be used, along with a set of reciprocity policies, as an incentive mechanism to encourage peers to contribute resources even when users are not actively consuming overlay services. This mechanism uses a decentralised credit network, is resilient to sybil attacks, and allows peers to achieve time and space deferred contribution reciprocity. Then, we present a novel, QoS-aware resource allocation model based on Vickrey auctions that uses *PledgeRoute* as a substrate. It acts as an incentive mechanism by providing efficient overlay construction, while at the same time allocating increasing service quality to those peers that contribute more to the network. The model is then applied to lag-sensitive chunk swarming, and some of its properties are explored for different peer delay distributions.

When considering QoS overlays deployed over the best-effort Internet, the quality received by a client cannot be adjudicated completely to either its serving peer or the intervening network between them. By drawing parallels between this situation and well-known *hidden action* situations in microeconomics, we propose a novel scheme to ensure adherence to advertised QoS levels. We then apply it to delay-sensitive chunk distribution overlays and present the optimal contract payments required, along with a method for QoS contract enforcement through reciprocative strategies. We also present a probabilistic model for application-layer delay as a function of the prevailing network conditions.

Finally, we address the incentives of managed overlays, and the prediction of their behaviour. We propose two novel models of multihoming managed overlay incentives in which overlays can freely allocate their traffic flows between different ISPs. One is obtained by optimising an overlay utility function with desired properties, while the other is designed for data-driven least-squares fitting of the cross elasticity of demand. This last model is then used to solve for ISP profit maximisation.

Acknowledgements

To Wallis.

Thank you for holding my hand in this amazingly beautiful universe.

Firstly, I would like to thank my mother, my grandmother, my grandfather and the rest of my family. I owe to you what I am, and what I aspire to be. I will always make you proud.

I would like to thank my supervisor, Miguel Rio, for letting my mind roam free, while still preventing me from wandering too far away. May sometime the world be just as free.

A big round of thanks to Richard Clegg, Dave Griffin and Eleni Mykoniati, for helping me become a better editor and academic writer, and and a more focused teamwork thinker.

To all my colleagues at UCL and Cambridge: thank you. The many discussions we have had over the years have helped me immensely in becoming a more balanced researcher.

Finally, the research in this thesis was possible through grants from the following funding bodies:

- The Mexican ministry of science and technology (CONACyT)
- The British Foreign Commonwealth Office through the Chevening programme.
- The Mexican ministry for education (SEP), through a complimentary scholarship.
- UCL, through a joint fellowship with CONACyT.
- The **PeerLive** project.

Contents

List of Figures

List of Tables

Part I

Introduction

1

Incentives in QoS Overlays: Problem Space and Summary of Contributions

Since its inception, the Internet has furthered an open philosophy based on highly intelligent endpoints communicating across a minimally intelligent, transparently interconnected collection of packet switching networks. Through the ubiquitous use of Internet Protocol (IP) and its associated core network protocols, the Internet acts as a substrate on which advanced application layer services can be deployed incrementally and with minimal intervention by network operators.

Initially, most Internet services assumed asymmetry in their protocol participants, with low-capacity clients requesting services from high-capacity, ISP owned and operated servers. However, as the cost for computation resources and access bandwidth decreased, and thus Internet connected hosts became increasingly powerful, new Internet services arose that shifted from away from this model and towards a much more user-centred one. First enterprises and then individuals started deploying their own servers, thus making the network edge more and more important for the provision of services. This continuing trend persists today in the shape of massively distributed, global-scale systems based on peer-to-peer protocols. The techniques on which these protocols are based have led to many innovations in distributed systems theory and practice, particularly in regards to content search, distribution, replication and caching (see Chapter 2 for an overview of some of these techniques).

Although peer-to-peer protocols provided great scalability and enabled the distribution of large files on a scale not seen before, they also made it clear that the task of designing and testing peer-to-peer overlays is far from complete. In addition to well-known (but yet unsolved) problems in distributed computing such as reliably synchronising state between peers (the *consensus* problem) and the analysis and reduction of messaging and computation overhead (computational and communications complexity), the design of reliable distributed incentive mechanisms remains an important open question in the peer-to-peer systems research agenda that has proved challenging in its analysis (for instance, from the point of view of Distributed, Algorithmic Mechanism Design (DAMD) [111, 304, 241]).

We are interested in the study of application layer overlays, in which sets of Internet hosts define

logical, application layer topologies that use the Internet as a substrate. We shall pay particular attention to the decentralised construction and maintenance of this overlays, since this will provide greater reliability under conditions of peer failure and churn (see Chapter 2 for perspectives on this issue). Finally, we further narrow down our interest to the study of *QoS overlays*, defined as decentralised application layer overlays which provide services which require predictable quality, even as the conditions of their underlying networks change.

This thesis will focus on the design of incentive mechanisms for these QoS-sensitive peer-to-peer networks, and in the modelling of peer preferences which is the basis for such a design. By relating incentive mechanisms to overlay resource allocation, we shall discuss their implications for various network design objectives and we shall present algorithms and models to solve the incentives problem in various scenarios. This will be achieved by drawing connections to social theory and economics, and to previous work in the modelling and analysis of peer-to-peer networks.

In addition, we will apply these general models to specific cases within the broad class of peer-to-peer overlays with QoS constraints, such as time sensitive media streaming systems.

1.1 The Incentives Problem and QoS Overlays

We now discuss, in an informal manner, the motivations behind the contributions presented in this thesis, leaving a detailed analysis of our contributions (and their role in the design of incentive mechanisms for QoS overlays) for Section 1.2. We start by considering the need for incentive mechanisms to control freeloading in peer-to-peer systems, which has been the most widely analysed aspect of incentive mechanism design for peer-to-peer networks (see Chapter 2 for an in-depth analysis of this issue).

Historically, freeloading was identified as a problem for peer-to-peer systems because many early peer-to-peer protocols allowed *non-excludable* [144] access to overlay resources[1]. This meant that, in these systems, all peer requests were served indistinguishably, regardless of the amount and quality of the contributions that the requesting peer had made to the overlay. Since peers could get access to overlay resources without contributing anything back, many chose not to contribute at all. Predictably, this externalisation of contribution costs led to widespread service degradation [167, 27].

Some early attempts to control freeloading focused on resource contribution accounting by eliminating peer anonymity. These were based upon the premise that, lacking stable peer identities, it was impossible for decentralised overlays to associate with each peer a measure of their contributions and refuse service if these were too low. Unfortunately, solving the identity identity management problem in peer-to-peer systems is challenging, particularly if one insists that proposed solutions are free from centralised components or trusted third parties. Furthermore, even if peers are bound to a particular identity at some point, it is an easy matter for them to bind to a new identity once the old one is no longer useful or has been labelled as a freeloader.

Thus, later proposals for the control of freeloading relied not on the control of peer identities, but on making resource consumption contingent upon resource contribution. The well-known *Tit-for-Tat* policy of BitTorrent falls in this category, as do many other schemes detailed in Sections 2.5, 2.6, 2.7, and 2.8. Alternatively, there has been work in protocols that make the use of multiple identities unprofitable [323, 262, 203] (see Section 3.2 for an in-depth analysis).

The practical motivation behind the aforemetioned work is clear: since the prices for computation resources and bandwidth have continued to go down, the potential gains in aggregating Internet end-hosts into large QoS overlays have become increasingly attractive. As the development of robust and reliable incentive mechanisms is a precondition for these gains to be achieved, the research community

[1] In economics, *excludable* goods are those for which access can be denied, particularly if payment has not been received. Conversely, *non-excludable* goods (such as, for instance, air) are those for which access cannot be denied.

has given this problem increasing attention (see Chapter 2 for an introduction to the very large research literature on the subject). One of the main contributions of this thesis will be the development of an indirect reciprocity mechanism to prevent freeloading. This mechanism relies only on self-certifying identities, and is resistant to many identity-based attacks without requiring any trusted third parties or certification authorities.

Although freeloading is a fundamental issue for peer-to-peer overlays, it is only a symptom of a much deeper problem: that Internet end-hosts are not under the administrative control of the overlay protocol designers. This means that, since it is usually impossible to force them to behave in any given way, they can impose their own preferred behaviour on the system. This happens because, in general, the only pre-requisite that peers need to satisfy in order to function within the peer-to-peer overlay is that they respect the relevant protocol definitions in their interactions with other peers; beyond that, they have absolute freedom concerning the software logic they execute locally.

Thus, in the same way as it happens in standard economic settings, even if a protocol is designed to optimise some system-wide measure of performance, peers can choose to behave differently in order to maximise their own utility, with no regard for the utilities of other peers. In this more general setting protocol design is further complicated, since the self-utility maximisation behaviour of the peers might not be compatible with the welfare-optimisation objectives of the system designer.

The aforementioned problem is the main challenge underlying the design of incentive mechanisms, which, formally, seek to align the utilities of the system and each individual peer participating in it (a shared objective with *mechanism design* techniques [160, 111, 241, 77, 232, 136, 304]). Thus, when engineering overlay network protocols for self-interested agents (and their associated incentive mechanisms), the protocol designer must define a set of protocol rules that include the desired outcome as an equilibrium, while at the same time, ensuring that individual or coalitional deviation is unprofitable. This is, in the general case, a very difficult problem (see, for instance, [174, 220] and Chapters 2 and 3 of [242]).

In most settings of interest to peer-to-peer designers, and in many models presented in this thesis, it is assumed that the only way that a peer can provide value to the overlay is through the contribution of some of its resources, and the only way that a peer can extract value from the overlay is by consuming the resources of other peers. Thus, the contribution and consumption of resources become the most important elements in defining peer utility, and an incentive mechanism becomes a set of rules that define the relationship between the contributions that a peer provides to the peer-to-peer overlay and the resources that it consumes from it. Thus, in this particular case, the problem of aligning selfish peer behaviour with desired protocol behaviour becomes one of *resource allocation*, and an incentive mechanism becomes a set of of rules that stipulate which peers can gain access to which resources, and under what circumstances.

If we consider the main design objectives of a peer-to-peer overlay, we see that it usually involves the the efficient distribution of content in a self-organising, scalable way. This is particularly important in the case of QoS overlays, whose resource allocation is driven by complex protocols in which the performance of the entire overlay becomes a function of which peers get service from which other peers, and the large-scale service distribution topology induced by this (see Section 2.3 for specific examples of how peer selection and resource allocation are adapted to a particular definition of service quality). Thus, the execution of efficient resource allocation functions is the explicit design objective (the *raison d'être*) of QoS overlays - which brings us to one of the central problems addressed in this thesis: as resource allocation is the fundamental design tool for QoS overlay protocol design, resource allocation rules related to the provision of incentives must coexist with those responsible for optimising the performance of the system.

Conceptually, we can see the design of an incentive mechanism as an optimisation problem, where the objective function is the alignment of peer and system incentives. On the other hand, the design of the peer-to-peer protocol itself can be imagined as an optimisation problem where the objective function is the performance of the overlay in the efficient transport and caching of content. These two objective functions might be very different indeed, leading to a conceptual model where overlay performance and incentive alignment need to be jointly optimised, with the appropriate tradeoff between the two being the main question to be answered in any particular proposed design.

In other words, since efficient and decentralised resource allocation algorithms are the basis for the design of QoS-overlay protocols, incentive mechanisms need to take their objectives and demands into account, providing a way for tradeoffs to be made between the provision of incentives and the optimisation of the peer-to-peer overlay. This will be denoted as *the incentives problem*, and proposing a particular algorithm for its solution will be another main contribution of this thesis.

In addition to the incentives problem, QoS overlays deployed over the best-effort Internet have specific characteristics that set them apart from other peer-to-peer systems. In particular, when the definition of a resource contribution is extended to include some notion of quality, a question arises on whether the service quality received by a peer is representative of the behaviour of the peer serving it - or just a consequence of the behaviour of the network between them. Thus, there is uncertainty on whether the effect of a given action by a certain peer is a consequence of its own decisions (up to and including deviation from the overlay network protocol) or a consequence of extrinsic factors, such as prevailing network congestion. The ramifications of this uncertainty in the context of incentive mechanism design are far-reaching, since it has an impact not only on the determination of how much value do the contributions of a given peer actually have for the peer-to-peer overlay, but also on the predictability and the quality guarantees that the overlay can ultimately achieve. Since the accurate modelling of peer contributions and their effect on the quality levels that the peer-to-peer system can deliver is central to the design of incentive mechanisms, another core contribution of this thesis is a possible solution to this problem.

Although a large proportion of this thesis will be concerned with the analysis of the incentives of individual peers, it is possible to apply the same modelling and analytical techniques to other situations where incentives tensions surface. One of these is the interaction between managed overlays and their underlying network connectivity providers (ISPs). In this case, however, our interest on incentives is not related to protocol deviation or freeloading. Instead, our main question will revolve around the predictive capabilities that an ISP could have on the incentives of a set of managed overlays that use its infrastructure to deliver QoS-sensitive services. Thus, rather than modelling the incentives of single peers and their alignment with the goals of a protocol designer, we postulate an incentives structure for the managed overlays, and then attempt to formulate models for their behaviour as a function of the behaviour of their underlying ISPs. We do this by proposing a utility function, and then obtaining the optimal managed overlay behaviour given the incentive structure of the ISPs, as expressed by a set of end-to-end prices. In particular, this utility optimisation framework can cast light on the load balancing and traffic flow allocation incentives of managed overlays, while at the same time providing indirect evidence that a particular economic framework provides ISPs with an incentive for the provisioning of high quality, low price links. The development of such a model, as well as its analytical solution, will be the fourth and final contribution presented in this thesis.

1.2 Desired Properties for QoS Overlay Incentive Models

As previously discussed, peer-to-peer systems for which there is no contribution accounting and an associated incentive mechanism in place can suffer from pervasive freeloading and widespread quality

degradation. In addition, the objectives of an incentive mechanism not always align with those of the request scheduling and distribution topology construction algorithms employed on a given peer-to-peer system. Again, a conceptual framework that explains this is simply that each of these two algorithms optimises different objective functions: in the first case, one related to ensuring fairness in the access to resources, and in the second case, one related to maximising some quality measure.

Thus, as previously discussed, the task of achieving a balance between provisioning incentives for participation and building efficient distribution systems (what we have called the *incentives problem*) takes the conceptual form of a multi-objective optimisation problem with complex constraints, usually with "tradeoff" solutions that embody some degree of arbitrary assumptions (usually in the definition of the peer and system utility functions). Thus, in presenting the contributions made in this thesis, it will be advantageous to define which were the engineering objectives and assumption that guided the model decisions presented in the next chapters.

The following objectives were the outcome of the literature review presented in Chapter 2. Essentially, when assessing the state of the art, we found that a large proportion of the research literature on incentive mechanisms failed to cleanly delineate the incentives problem, instead focusing on either participation incentives or topology construction. Customarily, this main focus is then extended with ad-hoc extensions to solve the incentives problem.

Furthermore, only a minority of studies focus on the provision of incentives for the specific case of QoS-enabled peer-to-peer overlays deployed over the best effort Internet. Thus, by exploring the application of traditional microeconomics techniques to guide the design of protocols that use QoS as a tool for incentives provisioning, we aim to identify fruitful areas for further research. We aim to capture this particular research inclination by establishing some traditional results from microeconomics as desirable engineering objectives, when applied to QoS-enabled overlays.

Our basic assumption is that the main goal of any incentive mechanism will be to elicit peer behaviour that leads to the continued availability of overlay resources. Thus, as the resource demands that a given peer imposes on the overlay grow, the incentive mechanism must ensure that its resource contributions grow proportionally.

Property 1 (Fundamental Property of Incentive Mechanisms)**.** *An incentive mechanism makes the level of access to overlay resources, including the service quality obtained from the system, contingent on the amount and quality of the resource contributions that peers have made to it.*

This means that one main objective of the incentive mechanisms to be presented is to account for resource contributions, and then use this information to drive quality and resource allocation decisions. We now present some other properties that we consider as desirable on a QoS overlay, and that are therefore present as motivating assumptions in the contributions that we summarise in Section 1.3.

1.2.1 Deferred Indirect Reciprocity

Although many incentive mechanisms have focused on the instantaneous reciprocation of overlay contributions (for instance, [84, 248] and [201]), we consider it advantageous to allow peers to decide when, and with which peers, will their contributions to the overlay be reciprocated. We consider systems that enable peers to use their contributions to obtain services from a set of peers different from those to which contributions were made, and at a different time.

One of the reasons why contributing to a given peer while obtaining services from a different one is useful involves user consumption patterns. In particular, it is advantageous for users (particularly those in asymmetric connections) to contribute over long periods of time (during the time the user is not actively using her computer, for instance), store these contributions, and consume them at a later time, at

a possibly much higher rate. This allows peers to consume resources at their download rate, even if they are unable to contribute to the overlay (upload) at this same rate.

An additional benefit of a scheme like this is that it allows peers to contribute resources to the system even if they are not interested in consuming those resources available at the time, thus reducing the need for a simultaneous coincidence of wants between peers that are capable of providing a service and those that require it. We state the following desirable property for our incentive mechanisms in light of the discussion above[2].

Property 2 (Deferred Indirect Reciprocity)**.** *The incentive mechanism is designed to allow peers to contribute services to a set of peers at a given time, and obtain resources corresponding to those contributions from a different set of peers at a different time, with no previous direct reciprocity requirement between them.*

1.2.2 Resistance to Identity Attacks and Untruthful Peers

The usual technique to achieve Property 2 is through the use of reputation systems (see Section 2.7), where peers implement extended recommendation networks. However, most reputation systems are vulnerable to *sybil*, *slander* and *whitewashing* attacks. While sybil attacks (the creation of arbitrary numbers of identities) and whitewashing attacks (the discarding of identities that have been labelled as malicious by other peers) depend on the very low cost of generating new identities [203], slander (lying regarding the contributions of other peers) depends on the capacity of peers to modify protocol messages as they forward them.

Since we do not assume a-priori that infrastructure is in place to aid in the solution of any of these problems, we state the following desirable properties for our incentive mechanisms.

Property 3 (Resistance to Identity Attacks)**.** *The incentive mechanism is designed to be resistant against sybil and whitewashing attacks, by ensuring that the profit that peers can obtain from assuming multiple identities is limited.*

Property 4 (Resistance to Untruthful Peers)**.** *The incentive mechanism is designed to be resistant against untruthful peers, by ensuring that peers are unable to profit from modifying or discarding the protocol messages sent by other peers.*

1.2.3 Decentralisation

Since it is impossible to ensure the continuous presence of every peer in the overlay, peer-to-peer systems need to be inherently resistant to peer churn, random peer failure and even denial of service attacks. This is much easier if single points of failure can be avoided.

We will aim to design resource allocation protocols that are as decentralised as possible, and thus, in general we will not assume that central control points are available. Furthermore, we note that one of the most important characteristics of a peer-to-peer network with regards to the development of incentive mechanisms is that each peer can potentially be under the control of a different administrative authority. This implies that federated approaches that rely on arrangements between administrative domains might be impractical in this context. We state the following desirable property in this regard.

Property 5 (Decentralisation)**.** *The incentive mechanism is designed to operate without need for centralised or federated infrastructure relying on agreements between administrative domains.*

[2]It is interesting to note that Property 2 does not require any specific indirect reciprocity scheme. We shall present an specific proposal based on arbitrarily long transitive contribution chains in Chapter 4. In practice, it may be that small transitive chains are enough for good enough performance under some circumstances [255].

From a system design point of view, we place the additional restriction that peers only keep track of the contributions that they have given or received, so that only the two peers involved in the event of providing a resource contribution (the peer who gives it and the peer who receives it) store a record for it. This will make system design much simpler, and eliminate much of the traffic that is necessitated by distributed peer banks [132, 307] or threshold cryptography techniques [328].

Property 6 (Private Storage)**.** *The incentive mechanism is designed so that peers store a record only of the contributions that they have given to other peers and received from other peers, not the contributions between third parties.*

1.2.4 Swarming and Overlay Topology Formation

We are interested in the development QoS-aware peer-to-peer protocols that use resource allocation both for the construction of efficient distribution overlays and to provide incentives for resource contribution. We approach these two problems simultaneously because, as has been discussed before, incentive mechanisms must take into account the resource allocation particulars of the overlay network for which they are being designed. We will call a decentralised resource allocation mechanism a *swarming protocol*, and now discuss some desirable properties that a it should have.

The first property that we address for the overlay construction protocol is that it should be *efficient*. This means that the system should be biased towards maintaining overlay associations between peers that give each other good quality service, and avoid maintaining overlay connections for which the quality is bad. Related to this, the system should provide peers with an incentive to shift their service requests from peers who experience high load to other peers who can provide the service, but that have lower utilisation levels. We call this property *load awareness*, and swarming algorithms having this property will be able to provide peers with hints on what the balance between the load and the capacity of each peer is, and thus, which are the peers with a higher probability for obtaining good quality service. Of course, we seek these properties to be achieved simultaneously with Property 1: the incentive mechanism must still provide peers with increasing benefits as their contributed resources increase.

Another desirable property under consideration is that the system should allow peers to estimate the benefit that an interaction with another peer might give them. This will allow peers to trade off contribution costs with service quality in an individualised fashion, allowing a great degree of flexibility in the system. We shall call this property *preference freedom*, and in practice it will mean that we do not require all peers to calculate their utility from the system in the same way - the system should allow each peer to have its own utility function. We now formalise these properties.

Property 7 (Efficiency)**.** *The system maximises the aggregate value that the peers obtain from it, by giving priority to those service requests that bring the most benefit to other peers.*

Property 8 (Load-Awareness)**.** *The cost of obtaining services from a peer increases as its usage level approaches its capacity.*

Property 9 (Preference Freedom)**.** *Each peer is able to define its own set of preferences over possible resource allocations. Equivalently, the benefit that a given peer obtains from the system is only decided by that peer.*

1.2.5 Adherence to Advertised Service Levels

The earliest peer-to-peer networks did not include the notion of multiple levels of quality of service, other that the ability (or lack thereof) to make use of the system at an essentially random level of quality. When, however, one considers peer-to-peer networks with QoS classes deployed over the best effort Internet, a *hidden action* problem surfaces. Essentially, it is impossible for the recipient of a service to discern

between the situation where a QoS contract was breached because the serving peer delivered degraded service itself, and an alternative situation where the serving peer delivered the service according to the QoS contract, but its quality was degraded by the network path between itself and the client.

Two independent properties are required to address this issue. These do not have the same level of generality as the previous ones, but will be nevertheless included as they are useful design objectives addressed by contributions in this thesis.

Property 10 (Network-Based Service Quality Model)**.** *The incentive mechanism includes a model of the influence of network conditions on the service quality received by a peer, as a function of the behaviour of its serving peer and the prevailing network conditions.*

Property 11 (Resistance to Hidden Action)**.** *The incentive mechanism provides an incentive to serving peers to deliver their advertised service quality accurately, even if non-compliance could be in theory be attributed to the prevailing network conditions.*

1.2.6 Incentive Tensions Between Managed Overlays and ISPs

Most game theoretical models of peer-to-peer overlays focus on the incentives tensions between peers (see, for instance, [141, 61, 35, 63] and Chapter 2), thus not including the ISP as an active player (interesting exceptions can be found in [31, 211] and the selfish routing literature). However, given the large amount of bandwidth that peer-to-peer systems can command, the ISPs have taken a much more active role in the management of peer-to-peer, spanning from techniques such as bandwidth capping or throttling [312, 322] to cooperative approaches such as P4P/ALTO [320, 252].

Although the modelling of the incentive dynamics of coexisting managed overlays with a single underlying ISP is interesting in its own right, we seek to model the preferences of overlays in the more general setting where they can choose to allocate their traffic matrix between a set of different ISPs. By considering such an "unbundled ISPs" model together with multiple competing P4P-enabled overlays, it becomes possible to consider a much wider array of preferences, thus achieving a more general model for the incentives of managed overlays.

We consider that a model of managed overlay preferences that allows ISPs to predict the traffic matrices that an overlay will demand as a given of a given price structure by itself and its competing ISPs would be of great help for network dimensioning, profit optimisation and traffic engineering. We present the following desirable properties on the basis of which such an overlay preference model can be scrutinised.

Property 12 (Flexibility)**.** *The preferences model for the managed overlay can be applied to a situation where there is only one overlay and multiple ISPs, multiple overlays and a single ISP, and both multiple overlays and ISPs.*

Property 13 (Ease of Computation)**.** *The preferences model for the managed overlay should be scalable, so that situations with thousands of network endpoints and hundreds of both ISPs and overlays can be considered.*

Property 14 (Parsimony)**.** *The preferences model for the managed overlay should make use of the minimum number of parameters.*

1.3 Summary of Contributions

We now turn to a summary of the main contributions found in this thesis, and relate them to the desirable incentive mechanism properties defined in Section 1.2. A condensed representation of this can be found in Figure 1.1, that shows the relationships between the chapters in this thesis, and the contributions that can be found in each one of them.

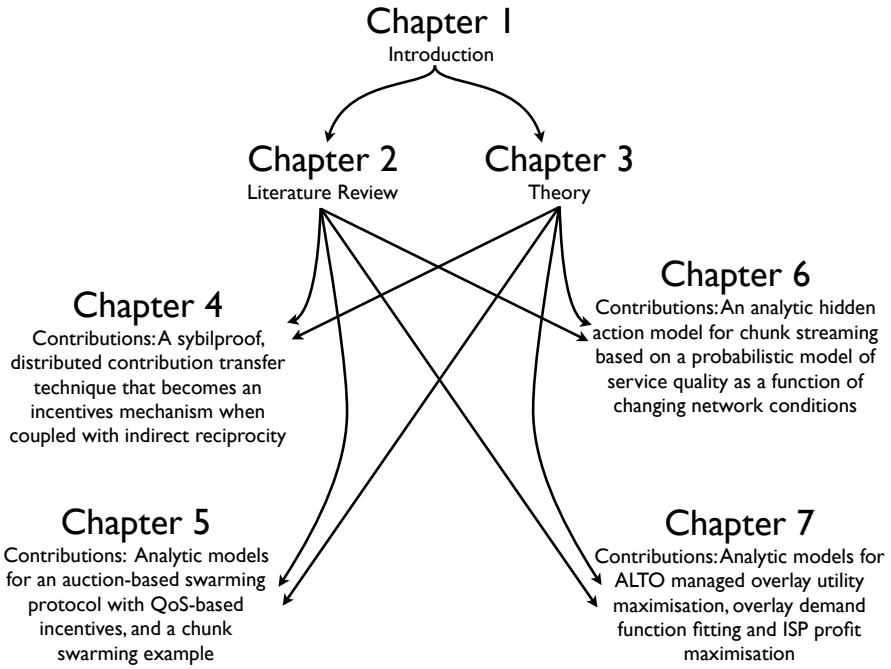


Figure 1.1: Relationships between the chapters in this thesis.

1.3.1 Identity, Accountability and Sybil Attacks

We directly address Properties 1, 2, 6, 3, 4 and 5 in Chapter 4. Essentially, we propose a robust contribution accounting system that allows indirect reciprocity without requiring peers to rely on possibly false third-party information about the contributions of other peers. The system relies in a *contribution transfer* operation based on path-wise maximum flow, and is designed to be *sybilproof* [71, 262]. This operation allows the transfer of previous contributions using source-routed messages over the contribution network of the peer-to-peer system.

The system uses self-certifying identities, is decentralised, bootstraps using reciprocative altruism and is designed to be resistant to sybil and whitewashing attacks. Since the system relies on decomposing indirect reciprocity as a routing problem, we present a biased sampling scheme to extract contribution subgraphs over which contributions can be transferred using self-avoiding, truncated random walks for topology discovery.

Our contributions in Chapter 4 are presented in three main parts. Firstly, we present the generalities of our accounting system (Section 4.3), our proposed probabilistic topology sampling algorithm (Section 4.4.1) and our contribution transfer protocol (Section 4.4.2). Secondly, we present our proposed algorithm to seed the contribution network by identifying trust cycles (Section 4.5). Thirdly, we describe our incentives system (Section 4.6). In Section 4.7 we present the attack resistance properties of our proposed scheme, and in Section 4.8 we present some evaluation results.

1.3.2 Overlay Performance Optimisation

In Chapter 5 we address Properties 7, 8 and 9. The main objective for this chapter is to present a general model for resource allocation and incentives in QoS overlays, and then apply it to one specific example. This is done by leaving the QoS definitions for the specific services in question open, to be defined through the use of a *valuation function* and a *quality mapping function*. The resource allocation model presented, based on multi-item Vickrey auctions, balances overlay optimisation with incentives by making access to resources contingent on service value to the receiving peer, thus giving preference to those overlay links that increase overall system utility the most, while at the same time giving peers with

greater contributions to the overlay better quality service. In this case, the resource allocation tension between the incentive mechanism and the maximisation of overlay utility is resolved by breaking resource allocation in two phases (*access* and *quality mapping*) and allowing each one of the aforementioned processes to drive one of these phases. The system uses Vickrey auctions to decide which requests to accept, a greedy utility maximisation heuristic to decide to which peers to send requests and how many requests to send, and a ranking procedure for the contribution-based assignment of service quality to service requests.

The structure of the Chapter 5 is as follows. In Section 5.1 we present the basic elements of an overlay optimisation technique based on multi-item Vickrey auctions. As part of this technique, in Section 5.2 we present closed form formulae for the estimation of parameters of interest such as the expectation of the number of service units that a peer will win and the expectation of its utility, taking into account both the benefit it obtains from the system and the cost it needs to pay for it. In Section 5.2.2 a ranking-based incentive mechanism is presented that operates in conjunction with the aforementioned auction-based overlay optimisation technique. In Section 5.3 we present a peer selection algorithm that proceeds by local maximisation, sending each additional bid to that peer that offers a larger utility for it. Finally, the framework presented in Sections 5.1, 5.2 and 5.3 is applied to a simple chunk swarming scenario, and some characteristic performance measures are presented for different peer delay distributions.

1.3.3 Non-observability of Peer Behaviour

The derivation of models that satisfy Properties 10 and 11, is the main contribution of Chapter 6. By taking a *hidden action* model usually used in microeconomics for the control of externalities in managerial contracts and adapting it to be used in an overlay network setting, we present the optimal payments that provide an incentive for the server peer to deliver its advertised service quality, even if failure to do so could be misinterpreted by the client as a consequence of fluctuating network conditions. We apply this model to a chunk swarming scenario, and as part of the optimal price calculation we present a statistical model for the quality outcomes of the client peer as a function of the behaviour of the server peer and a model for the effect that the network has on service quality.

We propose a *principal-agent* model for hidden action that gives server peers sufficient incentives to meet their advertised effort levels, without client peers having to decide for each transaction whether the outcome was due to server behaviour or network conditions. This allows peers to draft contracts that provide incentives for truthful revelation of QoS capabilities, and to have predictable transaction quality.

The structure of Chapter 6 is as follows. In Section 6.2 we approach the general problem of hidden action in QoS overlays, and present the optimal differentiated payments that a client must offer to provide a server with the incentive to deliver its advertised level of service. In Section 6.3 we apply the model to a delay-sensitive chunk distribution overlay for media streaming, and show how the model parameters can be defined in terms of observable network measurements.

1.3.4 Price, Profit and Managed Overlays

We address the development of managed overlay incentives models in Chapter 7, where we present two alternative models for overlay preferences and consider them in light of Properties 12, 13 and 14. These models can be used by an ISP to optimise its own operation, since they allow the ISP to anticipate overlay bandwidth demand as a function of the prices and overlay link qualities that it and its competing ISPs offer. The first model is based on postulating a utility function that has desirable characteristics, and then solving it to obtain the maximal overlay network utility in a constrained optimisation problem (we consider a budget constraint).

Since it is not always possible to measure the parameters of the previous model reliably, a second model which is entirely based on curve fitting is also presented, and then used as a restriction for a

profit-maximisation problem for the ISP. This model is based on the estimation of the cross elasticity of demand between overlay flows using least-squares regression.

The structure of Chapter 7 is as follows. In Section 7.1 discuss the situation to which we aim to model: multiple overlays can compete with each other, and can choose from between various ISPs to route traffic between their participating sites. In Section 7.2, a model of an utility-optimising managed overlay is presented for the situation described above, and solved in closed form, both for an ESP with infinite wealth and an ESP[3] with a binding budget constraint. In Section 7.3 a model based on Cobb-Douglas constant elasticity utilities is presented to fit the overlay demand function experimentally, and this model is later used in Section 7.3.3 to present a solved model for a profit-maximising ISP.

[3] As defined in Chapter 7, an ESP is an Edge Service Provider - a managed overlay.

2

An Overview of Solutions to the Incentives Problem

2.1 Why Incentive Mechanisms for Peer-to-Peer Systems?

The Internet, as an open platform for communications services innovation, has become wildly successful. As most of its intelligence is delegated to the user-controlled endpoints, it can quickly and flexibly use a set of core communications services, like network layer routing, to implement all kinds of application layer services. However, this same openness implies that some of the most important protocol software of the Internet resides *outside* the network, like the TCP implementations on the diverse operating systems present in Internet-attached hosts. Thus, much of the behaviour of the Internet is under the control of end users. This is particularly true with respect to peer-to-peer technologies: even though they are limited in their expressive capability by the services that their underlying Internet provides, they still have great freedom to interact with each other in almost arbitrary ways.

If overlay peers in the Internet could be considered *obedient*, they could be easily designed to maximise the global utility of the system: the greatest good to the greatest number of peers, one may say [48]. However, this might not lead to an equilibrium state where each agent is "content" with the cost-benefit it extracts from the system. Thus, if one considers *strategic* peers that will adapt to extract as much benefit as possible from the system while at the same time attempting to minimise or at least control their contribution costs, protocol design becomes more difficult - in essence, an exercise in the prediction and control of the emergent properties of a complex dynamic system.

There are two fundamental instances in which these kinds of problems have been studied in the broader context of human history: the understanding of the production, distribution and consumption of goods and services (i.e. *economics*) and the structure and stability of human societies (*sociology* and *anthropology*). In both cases, human beings are considered autonomous agents with volition: capacity to exercise will, choice, and decision, but subject to motivation and coercion.

In [86], Coleman makes this distinction explicit:

> *There are two broad intellectual streams in the description and explanation of social action. One, characteristic of the work of most sociologists, sees the actor as socialised and action as governed by* ***social norms****,* ***rules****, and* ***obligations****. The principal virtues of this intellectual stream lie in its ability to describe action in social context and to explain the way action is shaped, constrained, and redirected by the social context. The other intellectual stream, characteristic of the work of most economists, sees the actor as having* ***goals*** *independently arrived at, as acting independently, and as wholly* ***self-interested****. Its principal virtue lies in having a principle of action, that of* ***maximising utility****.*
>
> (quote from [86]; emphasis added)

The analysis of strategic behaviour in complex systems has also been approached by biology, as it might be argued that ecosystems have important commonalities with peer-to-peer systems. The non-random survival of individual traits acted to ensure that those genomes that equipped organisms with more successful strategies (those more capable to protect themselves, procure food or shelter, avoid detection or ensure the survival of their young) were selected over the rest. Evolution by natural selection is, in a way, a model of how to find utility-maximising strategies, as measured by survival and reproduction probabilities.

The parallel between human societies and peer-to-peer systems with strategic agents is clear as well. Game theoretical models of political and economic situations routinely model humans as utility maximising autonomous agents. Humans will seek to maximise their utilities, and sometimes this will imply the so-called *price of anarchy*: the optimality gap between a global social optimum and the aggregation of local optima where each agent maximises its own utility, taking the actions of other agents as given.

Thus, protocol design for strategic agents has much to learn from biology, economics and sociology. Unsurprisingly, some of the problems that had already received attention in these disciplines in the past have resurfaced in the context of peer-to-peer systems. Take, for instance, freeloading on so-called first generation peer-to-peer networks. In his classic article, *The Tragedy of the Commons* [158], Hardin analyses the inefficiencies of free access and unrestricted demand on shared resources. Hardin argues that private utility maximisation by each one of the peers might drastically reduce the global utility of the system, eventually decreasing every peer's utility. This can be clearly seen in a file sharing, voluntary swarming peer-to-peer file sharing system [167, 27][1].

Usually the *Tragedy of the Commons* is used as an argument for centralised economic control, as it shows that systems of agents guided by private utility maximisation can converge to very suboptimal outcomes (which are instances of *market failure*), and thus need external guidance in the form of taxation incentives: the actions of self-interested individuals do not promote the public good.

However, implementing a trustworthy central authority capable of performing this taxation in a peer-to-peer system is sometimes not only difficult, but counterproductive: the reliability, scalability and manageability gains stemming from a peer-to-peer distributed implementation are severely impacted by the introduction of single points of failure, such as this central policing authority[2]. Thus, the idea is to engineer behavioural patterns for peers that allow them to guide each other to an economically viable, sustainable equilibrium that has appropriate fairness and efficiency properties, without the need of a central authority. In some sense, we seek to engineer software agents that interact freely while implementing Hardin's solution to the *Tragedy of the Commons*: mutual coercion, mutually agreed upon. How to accomplish this and the situations under which it is feasible are part of the objective of this study.

[1] An example would be eMule[7], but with no credit buildup.

[2] This limitation is of interest even in the case of *offline* systems where punishment is not immediate, since in those cases greater overlay resistance to auditing point failures can only be accommodated in detriment to the capacity of the overlay to dynamically respond to peer behaviour. Of course, this in itself can be detrimental to the stability of the incentives mechanism.

2.1.1 Peer-to-Peer Resource Allocation with Strategic Peers

From the earlier discussion, it is clear that the control of externalities is a prerequisite for the implementation of Internet-scale, sustainable peer-to-peer systems. One way achieving this is to allocate system resources preferentially to those peers that contribute more to its operation. Thus, we start our study of peer-to-peer incentive mechanisms by considering distributed resource allocation mechanisms, so that we can later on guide their operation as indicated by the peer resource contributions.

In general, resource allocation architectures can be classified as centralised, decentralised, hierarchical, and hybrid. Centralised schemes usually involve complex combinatorial optimisation algorithms (see [72] and references therein). As a consequence of scalability and reliability issues, centralised resource allocation has not been widely used in peer-to-peer systems. Decentralised schemes, on the other hand, usually require higher communication overheads [289], but these can usually be accommodated with the current technology. Thus, most of the commercially successful peer-to-peer systems use this sort of resource allocation strategy. Hierarchical systems have a rich tradition in the classical distributed computing literature, as they approach the tight performance bounds of centralised systems while achieving much greater scalability. Some examples can be found in Web caching architectures [331], grid computing [62], fair packet queueing [46] and file sharing in peer-to-peer systems [195]. **Napster**, based on a hybrid resource allocation system, uses a large cluster of dedicated servers to index the files that are being shared by active peers at each moment in time. When a peer searches for a given file, it queries these central servers through a set of dedicated connections. The servers then cooperate in a distributed fashion to process the query and return a list of matching files, and their locations, to the peer. After receiving the results, the peer may then select one or more files and locations from this list and initiate file exchanges directly with other peers. Another instance of hybrid resource allocation is **BitTorrent**, that relies on centralised *trackers* to point peers to other peers in a swarm, but from then on, uses a fully decentralised data transfer paradigm.

A common underlying theme to many of the approaches above is that they assume the peers to be either *obedient* (always following the prescribed protocols) or *adversarial* (seeking to actively disrupt the protocol). *Incentive-based* mechanisms take a different approach, by considering peers as rational protocol entities that will behave *strategically*[3](see [241] and [111]). We cannot simply expect each Internet host to faithfully follow the designed protocols or algorithms. It is more reasonable to expect that each computer might try to manipulate them for its owners' benefit, and factor this in as part of the engineering process.

2.1.2 Freeloading

Freeloading in peer-to-peer networks is a well-known problem (see [27, 277, 100, 276, 114, 181, 186, 209] and references therein, and [114] for a review of the literature from the standpoint of accountability and externality elimination).

Freeloading has been problematic in the context of peer-to-peer networks because many of them have been built around the assumption that peers would derive enough benefit from the system to willingly donate their resources for the common good. However, if a peer is able to enjoy the resources of the peer-to-peer system without giving anything in return, the cost of its own contribution could be bypassed to increase its utility. Thus, for each peer individually, the rational choice is to *freeload*, and be supported by other contributing peers. If, however, an increasing number of peers take this stance, a shrinking amount of resources will be shared among a growing number of users, until an eventual system collapse when the costs of the users donating their resources outweigh the benefits they reap from other donors. This is an equilibrium condition: once a critical level of freeloaders has made the

[3]Strategic peers pursue their own purpose, and will use any resources available to them to increase their utilities.

system unusable, there is no incentive to donate resources until the freeloaders leave the system. This freeloading behaviour has been confirmed for many peer-to-peer systems, including Napster[277] and Gnutella[27, 167].

There are many reasons why strategic peers would choose not to contribute resources to the system. One such reason is the performance impact that its own downloads might suffer as a result of its uploads. In [115], Feldman et al. explore the interference of uploads with download TCP acknowledgements as a disincentive for sharing. As upload bandwidth utilisation increases, the queueing latency of TCP acknowledgements increases, and thus the RTT for the connection as well. This decreases the download throughput, acting as a disincentive for the peer. Feldman et al. derive simple models for latency on source and destination dominated congestion scenarios, and conclude that the cost of sharing increases with the average level of sharing, and thus with the general performance of the system. This would imply that the prioritisation of acknowledgements in the peer's upload would significantly increase their utility in large peer-to-peer systems. However, in general terms, there is a disincentive for users to devote their entire upload capacity for the benefit of the peer-to-peer overlay; that is the reason why a vast number of peer-to-peer client users choose to *rate-limit* their peer-to-peer uploads.

2.2 A Taxonomy of Peer-to-Peer Systems

Peer-to-peer systems are conventionally defined as Internet-wide distributed systems formed by the pooling of resources of large numbers of low-cost, off-the-shelf systems, each under its own administrative control (usually individuals with basic computation and Internet connectivity resources). Although they can command formidable resources, due to to the sheer number of members that comprise them, they incur significant communication and processing costs in doing so. Reducing the inefficiencies related to their distributed resource allocation protocols is still an ongoing research problem.

The high degree of decentralisation of peer-to-peer overlays allows them to achieve very high levels of scalability at very low cost, but demands advanced search and content transfer protocols in order to achieve sufficiently high levels of efficiency. Because of this, many peer-to-peer system designers opt for hybrid architectures where some functionality is centralised.

In terms of the content dependency of their overlay topologies, peer-to-peer systems can be either *structured* or *unstructured*. Structured overlays are characterised by their strict control of the overlay topology, usually following the keyspace structure of an underlying DHT (Distributed Hash Table) which defines where media content is stored (this reduces content search to a routing problem). On the other hand, unstructured overlays use adaptive routing and flooding algorithms to implement topology-independent functionality.

The design of high-performance overlays for delay and QoS-sensitive content has been a very active area of research in the last 10 years [208]. In general, research has moved from a *tree-based* [98, 325, 42, 166, 175] to *multi-tree* based architectures [65, 66, 182, 247], and lately to *mesh*-based ones [75, 327, 37, 214, 300]. The *mesh* and *multi-tree* architectures have been compared [215], with the *mesh* systems exhibiting superior performance when evaluated under conditions of continuous churn. This happens, predominantly, due to the static mapping of content to a particular tree, and due to the definition of each peer as an internal node in one tree and as a leaf in all the others.

A further distinction is usually made between systems in which peers pre-arrange with their downstream peers which data subsets will be forwarded (*push* systems, such as [282, 42, 281, 65, 66, 166, 247]) and those where peers explicitly request data subsets from their upstream peers (*pull* systems, such as [327, 163, 253, 213]). The use of features from *push* systems to alleviate performance bottlenecks in *pull* systems has created a new class of peer-to-peer streaming algorithms, known as *push-pull* or *hybrid* algorithms (see [300, 215, 326, 194]). These algorithms combine deterministic topology formation and

chunk scheduling (the *push* elements) with arbitrary chunk negotiation and explicit requests (the *pull* elements) in order to achieve highly efficient swarming with low delay and a small number of repeated or delayed chunks.

There is an evolving drive towards the design of locality-aware peer-to-peer overlays (see, for instance, [50, 67, 271, 270, 64]). One particular solution to this issue that leverages advanced mathematical techniques is the deployment of *synthetic coordinate* systems, such as [94, 280, 200, 239].

A necessarily brief overview of peer-to-peer overlays follows, focusing in specific examples rather than an exhaustive enumeration (the reader is pointed to the many surveys on the area, such as [210] and [208]).

2.2.1 Structured Peer-to-Peer Techniques

In structured peer-to-peer systems, each data object is stored in a peer (or set of peers) in a deterministic fashion according to a unique *key* that resides in the same space as the unique peer identifiers (*peer-IDs*). The underlying DHT implements deterministic storage and retrieval operations: the *key* of the data object defines the *ID* of the peer where it will be stored, and the overlay route that searches for such a data object will follow. Thus, all peers must maintain routing tables that allow them to forward data object queries efficiently, so that they are received by the peer whose peer-ID is responsible for the keyspace slice to which the requested key belongs.

Where the various structured peer-to-peer systems differ is in the construction of the routing tables and the topology of the underlying keyspace. As there is usually no correlation between the topology of the keyspace and the delay characteristics of the Internet between any two peers on an structured overlay, the routing and search delay for these systems might be large when compared to a centralised data repository. However, their distributed nature makes them much more scalable and resilient to targeted attacks.

One early example of a DHT with some degree of locality awareness is CAN [260], a structured overlay whose keyspace is organised as a d-dimensional multitorus. Each peer is responsible for the management of a slice of object IDs (its *zone*), and routes object queries using greedy forwarding to the peer closest to the key. The system uses DNS for bootstrapping, and soft state coordinate maintenance messages for churn management.

In [329], Zhao et al. present Tapestry, an overlay system where a circular keyspace is organised as a *Plaxton mesh*: a static architecture for peer-to-peer search where peers route queries by forwarding them to the node whose ID is numerically closest to the destination of the query. The tables to achieve this have multiple levels, one for each digit of the key. Similarly to Chord, the number of hops that a query needs to reach its destination grows logarithmically with the number of peers.

Chord [231] uses consistent hashing techniques [188] to distribute subsets of the keyspace amongst a changing number of peers, so that the keyspace under the responsibility of each peer is approximately equal in size. The keyspace in Chord is the set of integers modulo $2m$ - a unidirectional circle formed by $2m$ keys. Peers in this keyspace then form an overlay network of logarithmically spaced *chords*, which ensures that a given query will reach its destination after $O(log(n))$ steps, where n is the number of peers in the overlay. Another system very similar to Chord is Pastry [64, 270], whose main contribution is the implementation of a routing overlay using a Chord-like DHT as a substrate. Since the weights for this routing overlay are supplied by the applications who use the system, any combination of metrics can be used - an obvious choice is the use of latency measurements to implement network locality awareness in content replication scenarios.

2.2.2 Unstructured Peer-to-Peer Techniques

In unstructured peer-to-peer networks there is no emphasis on prescribing any network topology to fit the structure of a keyspace. Thus, there is no deterministic way to ensure, ahead of time, the overlay route that a request must take in order to be received by the peer which might answer it. To make up for this, peers use optimised flooding techniques in conjunction with routing tables linking data object identifiers and "next-hop" peer identifiers. Thus, unstructured systems have markedly different performance as the replication of the data objects increases: popular objects are more densely replicated, and thus found quickly and efficiently, while unpopular objects are found either slowly or not at all. A by product of this independence between the position of a peer on the overlay topology and the keys of the data objects under its control implies that queries for unpopular data objects could be received by a large proportion of the peers in the overlay, and thus the routing and replication of flooded queries can place high computation and bandwidth demands on the peers. On the other hand, unstructured systems can be designed to incorporate network proximity in a natural way, and are inherently more robust against peer churn.

One of the earliest peer-to-peer systems, *Gnutella*, uses unstructured overlay techniques to implement distributed data object location capabilities over a flat peer topology[4]. Thus, it has no control over data object placement or overlay construction. Data object search is resolved using query flooding, limited by TTL. When peers receive queries, they check for matches with the data objects under their control (when necessary, data objects are assigned arbitrary IDs) and respond to the query originators if a match is found. Bootstrapping is accomplished with a set of infrastructure peers.

Another unstructured overlay is BitTorrent [20, 21]. While Gnutella provided distributed content location with point to point transfers, BitTorrent was developed to provide massively distributed file transfers (*swarming*), relying on centralised systems for peer location (via servers called *trackers*) and for content search and indexing, through off-protocol websites such as [13, 15].

BitTorrent is designed for the distribution of large files. The metadata related to a single file is stored in a *torrent* file, which includes the URL of a tracker. The tracker uses a simple HTTP-based protocol to maintain and communicate information on the all the peers participating in the torrent. After obtaining a peer list from the tracker, the peers can contact other peers directly for data transfer.

In BitTorrent data objects (files) are subdivided in constant length *chunks*, that are individually transferred through the overlay. When peers finish downloading a given chunk, they announce its availability to their peers, so that they can request it. Using these announcements and partial maps downloaded when initially contacting their peers, peers can build fragment chunk availability maps.

Data exchange in BitTorrent proceeds in 10 to 30 second window intervals. During each interval, peers measure the throughput obtained from their peers, and refuse to upload to those peers whose performance is unsatisfactory (this is called *choking*). In this way, peers use their own upload bandwidth as a resource to obtain as high download bandwidth as possible (details on this application of *Tit-for-Tat* as an incentive mechanism can be found in Section 2.6.1).

2.3 Swarming Techniques for Streaming Overlays

2.3.1 *Push* Peer-to-Peer Streaming Systems

In a push peer-to-peer streaming system each peer agrees on a long-term contract with its "upstream" peers, so that they will continue to forward streaming content continuously, with no re-negotiation or re-requesting needed. Thus, after a relationship between peers is defined, it will remain stable for a continued period of time. The content to be streamed is defined at the moment of contract creation, and does not need to be re-negotiated.

[4] Version 0.6 of the protocol does include *superpeers*, but in the original implementation all peers (*servents*) were identical.

Early systems for peer-to-peer streaming were based in the concept of *chaining* [282], where a centralised server offloads some of the streaming upload effort to downloading clients which also do it in a recursive fashion, building a long streaming chain. Although chaining is simple, it suffers from very high end-to-ed delay (in fact, near the maximum possible) and very small resilience towards churn or QoS instabilities in the distribution chain.

To reduce the playback delay from the stream source and increase efficiency, techniques were developed based on the construction and maintenance of an Application Layer Multicast (ALM) distribution tree. For instance, in [42, 281] Sherwood et al. present a streaming system (NICE) where a distribution tree is built by aggregating peers into clusters, which then are represented by a *cluster head*. The cluster heads aggregate themselves into clusters as well, which then elect a cluster head. This process continues until every peer is reachable by every other peer through a path composed uniquely of cluster heads. Although NICE had desirable delay, stretch and stress properties[5], it suffered from lack of resilience against churn and peer heterogeneity, which could lead to unbalanced tree constructions and thus unpredictable end-to-end QoS.

A direct way to improve the performance of a tree-based approach is to use, instead of a single application level multicast tree, a number of trees that have disjoint topologies. A system of this kind is SplitStream [65, 66], where a stream is divided in substreams (*stripes*) that are carried over independent application layer multicast trees. In order to prevent correlated failure of many trees given a single peer failure or churn event, SplitStream requires every peer to be an inner node in just one tree, remaining a leaf in all the rest. By applying this idea to the transport of video streams, CoopNet [247] delivers greater resilience at the cost of increased delay and signalling overhead (see Section 2.8 for a more complete description).

2.3.2 *Pull* and *Hybrid* Peer-to-Peer Streaming Systems

In a pull peer-to-peer streaming system, peers only agree to very short contracts - of possibly a single chunk in length. Thus, contracts need to be continuously re-negotiated in case of long interactions. The typical example of this is per-chunk requesting, where peers will only forward chunks if the "downstream" peer explicitly requests it. Pull architectures require much greater signalling overheads (possibly including increased delay), but are more resilient to peer churn and erratic peer behaviour, and more responsive to network congestion and loss events. Another benefit of pull architectures is that they can be naturally applied to arbitrary overlay topologies using gossip (epidemic) algorithms [108]. Thus, the pull architecture is favoured by *mesh* systems like PROMISE [163] or PULSE [253]. Another example is CoolStreaming [327], where Zhang et al. describe a peer-to-peer streaming system in which peers begin with a random set of neighbours from which they can request chunks, and progressively refine it depending on the QoS that they experience. Further, they present a heuristic scheduling algorithm to maximise the QoS that a peer can obtain from its neighbours, and algorithms to deal with node churn.

Recently, *hybrid* systems have been developed that take advantage of the benefits provided by both the push and pull architectures. In [326], Zhang et al. describe a mesh-based system where an unstructured overlay is formed using a gossip-based protocol, which is then used to implement a push-pull streaming mechanism. After an initial negotiation, sending peers begin pushing chunks to receiving peers, until they decide to re-negotiate their relationship (possibly with a different sending peer).

[5] *Stretch*, from [42], is defined as the ratio between the distance between two peers over the overlay tree and its distance over the physical network topology. *Stress*, from [166], is defined as the average number of identical copies of a packet that are carried by a given physical link.

2.4 Network Locality Awareness

2.4.1 Application Layer Anycast

Although there has been some research on anycast services at the network layer [41, 189], the difficulties in establishing such a service have pointed researchers to application level solutions. In [324] and [110],the authors discuss anycast resolvers similar to those currently used by DNS. In the application layer, Content Distribution Networks (e.g. [2]) can be used to to implement locality-aware DNS resolution. However, they rely on a network of proprietary servers and have shown widely varying efficiency [291].

The Internet Indirection Infrastructure (i3) [288] proposes a generic indirection mechanism supporting unicast, multicast, anycast and mobility. In i3, group membership is not scalable, since a single node maintains all members of a group. To support large groups, i3 uses a hierarchy of triggers that have to explicitly constructed and balanced.

Many anycast systems based on peer-to-peer techniques have been proposed. In [67], Castro et al. propose an anycast system to build and maintain application-level multicast trees for group management. The system is based on the publish-subscribe functionality of Scribe[271], which itself uses Pastry as an overlay substrate. As Pastry [270, 64] tries to minimise network delay stretch by using a scalar proximity metric (like number of IP hops) and Scribe enables the formation of publish/subscribe groups, the system in [67] is capable of performing anycast through Pastry using Scribe groups. Thus, the system is designed to respond to queries with the location of a resource using a proximity aware metric, but not necessarily to return the closest resource replica as measured in the proximity metric.

By framing the anycast problem as a constrained search problem, techniques based on content-aware query routing become applicable. In [265] Rhea and Kubiatowicz propose an algorithm based on *attenuated Bloom filters* to enhance the performance of Internet-scale location mechanisms. It uses Tapestry [329] as a locality-aware DHT, and is thus capable of finding replicated content that is proximity-metric "close" to any given query site. The system, however, might be insensitive to performance penalties on congested or long-distance links. By using approximation heuristics to the well known *Steiner Tree Problem*, Majumder et al. present a *publish/subscribe* routing technique [217] that could be used to build locality-aware anycast system within a polylogarithmic factor of the optimum.

2.4.2 Network Coordinate Systems

Synthetic coordinate systems associate a coordinate with each peer in an overlay network, in such a way that the distance between the coordinates is a good estimate of some network property measured between the peers, predominantly *round trip time* (RTT). This can be achieved efficiently by using a limited set of end-to-end measurements to extrapolate those distances between peers that were not explicitly measured. Thus, synthetic coordinate systems use a limited set measurements to model the structural properties of the Internet, and then use this model to predict end-to-end properties (such as RTT) between arbitrary peers.

The first step in the operation of a network coordinates system is generating a *sampled distance graph*, where links between peers represent distance measurements. Of course, there is no requirement on this graph to contain all peers in the Internet, nor for this graph to be complete - the point of this techniques is to rely in a small number of measurements to fit a distance model, which is then used as an approximation. This fitting process can be understood as metrically *embedding* the sampled distance graph onto a space that integrates some of the structural properties of the Internet. Examples of these include an standard Euclidean space [90], an Euclidean space augmented with a purely additive coordinate [94] or a hyperbolic space [280]. The embedding process can be viewed as an error minimisation procedure where peers are positioned in the space in such a way that the cumulative difference between

the measurements and the embedded distances is minimised. Once this embedding has been done, and to the extent that the embedding space faithfully recovers the structure of the Internet for the measure in question, geodesic distances over this space are good predictors of the actual end-to-end properties of the Internet [200].

2.5 Incentives and Resource Allocation - Markets

The use of markets to solve resource allocation problems in massively distributed systems is by no means a new idea [284], and it has been widely addressed in the literature. However, when compared to other algorithm design methodologies, it has failed to spawn a large body of computer science research or to achieve widespread distribution in end user applications.

As bandwidth and computing power become cheaper, many Internet connected computer systems are experiencing a large surplus of bandwidth and processing resources, so that aggregated together, a peer-to-peer system can harness a very large aggregated capacity. However, resource coordination at Internet-wide levels can incur very large overheads. Market economies [284] have been proposed to address this issue in the light of the following recent changes in the Internet landscape:

1. *Flashcrowds*: Even though there is an increasing amount of idle resources in end hosts, episodes of enormously increased demand have been shown to require large over-provisioning in order to maintain adequate quality of service (QoS) levels. This is compounded by the fact that, in many situations, flashcrowds are essentially indistinguishable from distributed denial-of-service attacks [88].

   Because in peer-to-peer systems server power scales directly with client power, peer-to-peer systems scale well with sudden increases in demand, allowing consistent QoS level to be presented at the application layer.

2. *Computational Power*: End hosts have increased enough in their computing power to be able to solve the combinatorial problems endogenous to pricing and best response function estimation.

Market-based solutions approach the resource management problem by assigning each peer the responsibility of procuring its own resources in an optimum way. Thus, each possible solution to a resource allocation problem will have a measure of how good it is for each peer, and each peer will adjust its own actions in order to maximise its utility given the actions of the other peers. In an environment with no scarcity, each peer would be able to access shared resources in such quantity and quality that its utility would be maximal. However, if resources are scarce, competition can be used as a tool for resource allocation. In general, mechanisms based on competition tend to operate well, producing allocations that are efficient even if agents are not completely capable of determining their best courses of action [138].

Thus, market resource allocation can be studied through competition, and the study of competition can be naturally approached from the perspective of game theory. Competition is then "a process of strategic decision-making under uncertainty" [224]. Market competition allows the integration of private information of all economic agents in a single information item (the resource price) that reveals its true value to the set of consumers. Additionally, this price estimation can be more accurate that the best estimation of any single economic agent in the group (for a very readable account on this, see [293]).

In one of the first papers addressing the use of economic models for computing resources allocation, [294], Ivan Sutherland describes a system in which different economic agents (research groups within a university) periodically receive an allocation of virtual currency (called *yen*) that cannot be saved beyond the period when it was received. This means that there is no incentive to hoard, and the economic actors focus its usage on trading and pricing. To avoid currency depletion and starvation, yen are automatically recharged periodically.

In [249], Papadimitriou states that "the Internet is unique among all computer systems in that it is built, operated, and used by a multitude of diverse economic interests, in varying relationships of collaboration and competition with each other." Thus, game theory and economics, with their fresh supply of techniques applicable to ensembles of autonomous agents with conflicting and competing interests, will be fundamental in the modelling, analysis and design of Internet-wide technologies. Some interesting research questions posited by Papadimitriou include the *Price of Anarchy* (the difference between the system-wide utility of an equilibrium attained without any central intervention and one obtained through central optimisation), the performance analysis of market-based equilibria, the use of mechanism design to ensure socially optimal equilibria, the role of externalities in privacy and security issues in the Internet, the development of market-driven clustering algorithms and the modelling of network topologies as equilibria of economic processes.

The idea that any consumption of peer-to-peer resources by a peer has a value, and can be therefore imbued with a price, is appealing. Thus, peers are provided with an incentive to devote resources to the peer-to-peer system (acting as servers) because they will be paid by consuming peers (acting as clients). This payment can be either in actual national currencies (pounds sterling, euros, dollars, etc) or in intra-system currency exchangeable by electronic goods or services. In any event, the currency units are assumed to have an intrinsic value, and game theoretical formalisms rely on the conversion of peer utility in currency terms for the determination of appropriate payments (see, for instance, [141]). There are a number of drawbacks for purely monetary systems, such as the need for a micropayment infrastructure, central usage accounting and contract dispute resolution, trust anchoring and delegation (in case of distributed bank functions), economic cycle control (inflation and deflation avoidance), valuation and pricing, and communication and computation overheads for market convergence.

These problems notwithstanding, market systems have been repeatedly proposed in order to solve the incentives and resource allocation problem in an scalable fashion. In [121] Ferguson et al. use economic models to coordinate the activities of strategic agents. A general framework is presented and applied to load balancing, resource allocation, flow control, data management and QoS enabled queueing. Another example is [159], where Hardy and Tribble propose including in the header of Layer 2 frames a field that describes the value of the frame, expressed in some fraction of an actual legal tender currency. When these packets are transmitted through inter-operator boundaries, the costs are accumulated and cleared periodically by using conventional electronic funds transfer. Thus, Hardy and Tribble proposes the idea of micro-payment based inter-domain billing, with reputation systems used for risk minimisation.

2.5.1 Early Market-based Systems - MojoNation

An early example of an actual peer-to-peer overlay implementation that included a market-based incentive mechanism is MojoNation, described in [225] by McCoy and Wilcox. MojoNation used centralised currency and credit windows to create an open market for file fragments, where fragment storage, tracking, search, caching and traffic relaying were all services paid for in virtual currency (called *mojo*). MojoNation attempted demand-informed caching in an attempt to achieve locality and performance gains, and erasure coding to improve availability. Additionally, peers in MojoNation implemented a reputation system that included QoS attributes and credit ratings. All these ideas notwithstanding, MojoNation was unable to establish itself in the peer-to-peer market, and some of the reasons for its failure are discussed in [316]. In particular, it was observed that users did not remain in the network for extended periods of time. The most common behaviour was to join the network, stay for less than an hour, and then leave to never return. In fact, more than 85% (and maybe up to 90%) of all users remained in the network for less than 24 hours. Additionally, of those peers who did participate in the network for more than a day, around 17% remained constantly online, while the rest connected only around 3 hours per day. Clearly,

peers had a tendency to be exceedingly unreliable, and strategies that relied on long-lived peers were difficult to implement[6].

However, this study was carried out from October 2000 to February 2001, when the behaviour and expectations of users were widely different than those of 2008: only around 9 million U.S. Internet users had broadband connectivity then [250], compared to 84 million in 2006 [164] and around 122 million in 2008 [165]. In addition, most Internet users in 2000 and 2001 experienced time-based billing, whereas most Internet users today pay flat rates, with possible download caps [89].

Another issue raised in [316] is that, at any given time, around 30% of all MojoNation peers were not Internet reachable due to NATs. An unknown number of additional users did have routable addresses, but were not reachable due to firewalls. This continues to be a problem [122, 151].

Another reason why MojoNation experienced network-wide disruption was the peer bootstrapping procedure. Initially, MojoNation used a centralised *introducer* server, which was easily overloaded when a flashcrowd hit the system after being advertised in a popular website[7]. Trying to solve this problem, MojoNation substituted its original central server by a virtualised set of redundant servers under centralised control (this solution was also implemented by FastTrack-based systems such as Kazaa, Grokster, iMesh and others, and also by Napster). Another way to set up the initial bootstrapping is to bundle with the software an original list of contacts that may or may not be under central control (like *Limewire*[12]), or requiring the user to obtain contact information for some peers manually, using out of system mechanisms such as the personal social network of each user (like *Freenet*[10]). Another design failure that Wilcox-O'Hearn discuss in [316] was the inappropriate use of erasure codes. The design objective behind their use was to increase data reliability, even in the presence of high peer churn and turnover: "...MojoNation used an erasure code to split the data into a set of shares such that any sufficiently large subset of the shares would suffice to rebuild the data" [316]. The system was designed in such a way that only half the peers that were present when the content was initially replicated were needed to retrieve it. However, if the number of peers is less than this critical number, no retrieval is possible - not even partial retrieval. Given the enormous turnover rates experienced in MojoNation, many of the peers that were online when some content was uploaded simply disappeared with the erasure code pieces on them, permanently harming data availability. Wilcox-O'Hearn identify the need to give preference to long lived peers to store file fragments, rather than treating newcomers and old peers as equals when deciding where to place the erasure code replicas (a dynamic, adaptive coding framework was not explored).

2.5.2 Market-based Systems in the Research Literature

Another currency-based system is KARMA [307]. In KARMA, each participant is assigned a single scalar, called its *karma*, that is at the same time a currency amount and a reputation rating. Whenever a peer contributes resources, its karma is increased (the peer is *paid*), and when it consumes resources its karma is reduced (the peer *pays*). The karma of a peer is controlled by a *bank set* elected through Pastry [270]. Each one of the member peers of the bank set of another given peer maintains its balance and its latest transactions. If no peers left the system, the amount of karma in the system would remain constant: KARMA does not allow peers to *create* wealth. Thus, when new peers join the peer-to-peer system and old peers leave, the amount of karma decreases, increasing the risk of starvation. To prevent this, there are periodic, network-wide signalling episodes where each peer learns the total amount of karma and the total number of users in the system, and adjusts the karma accounts that they control to compensate

[6] In a 2005 study [256], Pouwelse et al. track the distribution of a large file in a BitTorrent swarm, and state that less than 4% of all BitTorrent peers remain in the swarm 10 hours after they have finished downloading the file. However, later studies [290, 311] have reported increasing numbers of stable peers.

[7] From [18]: The Slashdot effect is the term given to the phenomenon of a popular website linking to a smaller site, causing the smaller site to slow down or even temporarily close due to the increased traffic.

for any surplus or missing karma. This is, from a practical perspective, of limited efficacy: any peer-to-peer operation that requires network-wide coordination can be very difficult to maintain successfully in practice.

Offline Karma [132] extends KARMA by relaxing its online transaction verification requirement on the bank set of each peer, therefore enabling offline transactions. Double spending of currency is no longer preventable, and thus the system is aimed at fraud detection and accountability. Offline Karma uses partial hash collisions to limit the rate of currency minting. Currency in this system is transferable by nested digital signing, in exactly the same way as iWat [273, 173] and SHARP [78, 125]. However, Offline Karma limits the growth in coin sizes (which was first encountered in the context of digital cash schemes [69]) by forcing each coin to be "re-minted" periodically during the lifetime of the coin. When a coin is re-minted double spending can be detected and its associated history, in the form of its nested signature sequence, removed (this might be beneficial for privacy reasons, as well). The peers responsible for re-minting are elected in a similar fashion to the bank set in KARMA: by proximity in the ID space of a DHT.

Another similar system, (PPay) [321], is a micro-payment scheme optimised for peer-to-peer applications. PPay seeks to eliminate the central bank bottleneck of common micro-payment systems by delegating some of its functions to the peers. Each peer purchases *raw coins* from a central bank and then is able to re-assign them to other users by digitally signing them. However, the coin growth problem of WAT tickets is avoided by implementing an online *coin reassignment* protocol, where the original holder of the raw coin is required to reassign the coin after each transaction. This means that online bank functions are taken by the peers themselves, and problems may arise if they are not continuously available to manage their reassigned coins. To this end, an *downtime protocol* is used. As payments are made offline, coin fraud cannot be prevented, but it can be detected and controlled. The system scalability is improved by using soft credit windows, which essentially implements micro-payment accounts for transaction amounts smaller that the minimum value of a coin.

One of the problems of market-based approaches is price determination, which can be addressed in various ways. In [313], Wang and Li propose a market-driven bandwidth allocation procedure for overlay networks with strategic peers. Peers trade bandwidth units (in the form of *one-hop flows*) for currency units with peers in their neighbourhood. Bandwidth providing and consuming peers and are chosen so that a utility function that considers upload utilisation, download utilisation, paid currency costs and earned currency income is maximised. Pricing is determined by using *reinforcement learning* techniques, so that the price charged for each unit of bandwidth maximises utility in the light of past system performance. The system uses the *Receiver Only Packet Pair* bandwidth estimation method [197] to determine bottleneck bandwidth and thus feed this into the local utility functions.

There have been attempts to create a generic framework for the practical implementation of currency based systems. In [302] a generic public-key based protocol for pairwise resource trading with strategic peers that define their own pricing policies, and where multiple competing currency systems exist, is defined. This protocol (LCP, Lightweight Currency Protocol) uses trust-based currencies in a similar spirit to SHARP [125] and iWAT [273]. These currencies are exchanged by means of four different primitives, each one implemented in a particular protocol message: **transferFundsRequest**, **transferFundsResponse**, **getDepositsRequest** and **getDepositsResponse**. Currency issuers expose a Web Services applications programming interface (API), by using WSDL (Web Services Definition Language [19]) to bind a public key with a domain name, and SOAP (Simple Object Access Protocol [17]) over HTTP (Hypertext Transfer Protocol [11]) to exchange protocol messages. This means that any peer can become a currency issuer, or use the currency of any other peer of its preference.

2.5.3 Systems based on Exchange/Barter Economies

The design of contribution accounting mechanisms can benefit from the analysis of bartering schemes for small communities, such local virtual currencies [173, 287] or peer-to-peer lending [192, 70]. The WAT system was conceived and launched in 2000, and it enables Japanese farmers to issue self-certified acknowledgements of personal debt (IOUs) that can be transferred along chains of trust, in such a way that there is no superior administration group controlling the relationships between any two ordinary participants [206]. Thus, in a way, the WAT system is a conversion mechanism between capital in currency form and social capital. In practice, the WAT system works by providing enhanced bartering possibilities for the users of the system, which is now spreading rapidly and widely in Japan (due to its government-independent nature, it is difficult to ascertain exactly what its scale of use is, and how many participants are involved in it). Its users tout it as an effective means of payment, which in addition creates and strengthens *cycles of trust* among groups of participants.

The operation of the system is simple. Any person can issue a WAT ticket, which becomes active by being signed[8]. The ticket represents a commitment on part of the issuer to present to the bearer, at redemption, certain goods (or to provide a given service). Each recipient of the ticket may pass it on to other economic actors by signing the ticket with his/her personal signature. Therefore, the ticket becomes a record of all the transactions that it has supported, and by navigating social trust chains, it transforms reputation and trustworthiness (social capital) into economic solvency.

In [36] the WAT system is complemented by integrating self-guarantees, in the form of legal tender currency. Thus, economic value on a given WAT system can be supplemented if necessary by tying its value to other currency systems (maybe even other complementary currency systems). This means that different currency systems can inter-operate with each other in a peer-to-peer fashion, by converting their monetary value into trust (operationally, this is similar to the currency exchange procedures for the Lightweight Currency Protocol [302]). Like KARMA (and conventional digital transferable cash [69]), iWat, the digital implementation of the WAT system [273], has the property that as a WAT ticket circulates the economy, it grows in size, as each participant peer adds its own digital signature. Another very similar system is SHARP [78, 125], an architecture for a generalised computational economy based on pairwise barter chains. It uses probabilistic assurance over resources[9] and self signed identities, and resource entitlement tickets that can be delegated through nested signing (this is identical to [273] and [173]).

Another way of allowing peers to create their own currency is by defining pairwise currencies: Each peer will have a currency to transact with every other peer. In this case, all transaction state information can be kept by the two interested parties (as opposed to [78, 125, 273, 173] where state is recorded in the tickets/coins themselves), and the system demands in terms of cryptographic protection are alleviated. One such system is SWIFT [295], where each peer maintains an account for each one of its market peers. Trading on these pairwise currencies amounts to subtracting or adding to the other peer's account. By explicitly constructing trading strategies that take into account altruism and risk preferences, and that track the contributions of each one of the peers to others, Tamilman et al. argue in [295] that participation in the system is advantageous for strategic peers, and that *moral hazard* (strategic deviation from protocol-defined operation - see Chapter 6 and Appendix A.4) is avoided.

Many of the trade mechanisms that have been detailed so far fail to differentiate between the actual tradeable resources of the peer (e.g. bandwidth) and the content that it possesses (e.g. media fragments). In [131], Garbacki et al. propose the formation of *alliance groups* to allow peers to contribute upload

[8] In Japan, a personal seal usually takes the place of a handwritten signature.

[9] This means that if a peer obtains a signed certificate entitling it to a given resource, there is no hard assurance that it will get it when the peer decides to make it valid. This is due to the fact that if the instantaneous rate of resource "cashing" exceeds the instantaneous rate available to the granting peer, it will be unable to comply.

bandwidth to other peers even if they do not possess any content that the other peer might be interested in. Basically, Garbacki et al. modify the BitTorrent protocol so that peers aggregate in trusted groups where peers download fragments on behalf of each other. Each peer in the group then becomes, for each particular swarm, either a *collector* that is interested in participating in the swarm, or part of a set of *allies* that are not interested in the swarm but help the collector to achieve high download rates by downloading fragments themselves (as instructed by the collector) and relaying them, in the expectation of reciprocity at a later time. They name their approach *Amortised Tit-for-Tat*, because the allies collect fragments on behalf the collector only out of a reciprocal altruism expectation. Since this reciprocative altruism expectation is not managed by their protocol, Garbacki et al. assume that alliance groups are formed only within a high-trust social network (e.g. between friends and relatives).

Another strength of [131] is that, by eliminating the assumption that sharing in peer-to-peer networks is done in terms of mutually useful content it decouples the problem of content search, distribution and replication from the incentives problem. In this case, peers use their resources to download content for which they have no interest, in order to "store" this contribution and use it later to download different content. Although the set of peers with which it will interact directly is the same (its *allies*), the peers with which the allies interact can be completely different.

2.5.4 Exchange Systems and Mechanism Design

There have been numerous utility-based analyses of the incentives problem with strategic peers from the point of view of differentiated resource allocation. The rationale for these is evident by defining the solution of the problem to be the set of protocol primitives (game theoretical *strategies*) such that the utility that a peer obtains from the system is a monotonically increasing function of its contributions to the overlay. Due to its obvious relationship to resource allocation problems in economics, trading has been repeatedly proposed as a viable strategy for the distributed solution of the incentives problem. For instance, in [61] game theory is used to model the interaction of strategic peers (in this case the traded commodity is disk space), and the possibility of using utility functions to drive the system to Nash equilibria with desirable properties. Buragohain et al. formulate the best response function dynamics as a discrete time dynamical system, and the characteristics of the system while converging to the Nash equilibrium is analysed by linearisation around the fixed point. A similar model is presented in [141], where Golle et al. apply a game theoretic model to a peer-to-peer file sharing application, much like Gnutella. Different strategies for sharing and downloading are defined, and utility functions are proposed. Golle et al. consider a *payment* term reminiscent of the payments used in mechanism design to align system and peer incentives [304, 241]. Then, Golle et al. estimate the payment that each peer must receive for sharing in order to make sharing at high levels a dominant strategy (see Chapter 3.1 for game theory definitions). However, the proposed system requires a centralised server and logging of all peer-to-peer transactions, making it of limited interested in a pure peer-to-peer context.

Another model based on economic models for the provisioning of public goods is [35], where Antoniadis et al. consider an allocation mechanism in which a shared resource is divided among a set of n peers in accordance with their utilities. Essentially, Antoniadis et al. propose mechanism-design inspired system based on a *system usage flat fee*. Participants in a Gnutella-like peer-to-peer file sharing system are forced to all share the same number of files, which is equivalent to imposing a constant and identical cost upon all participants. Those peers with positive utility participate in the system, and those with negative utilities do not, implementing the rationality system participation restriction in a natural way. The authors extend the model to consider files with different popularity ratings, and consequently, entailing differing costs to the peer hosting them.

The analytic model proposed in [35] describes a simplified scheme in which every peer i uses the same utility function $u(Q)$, where Q is the quantity of common good produced, but each one of them

multiplies it by a scalar θ_i so that $u_i = \theta_i u(Q)$. Therefore, reporting u_i to a central authority only involves revealing θ_i. Each peer declares its parameter θ_i to a central authority which sets the level of production Q, determines which peers can use the common good, and the price each one should pay for access to it. The mechanism is designed to maximise the expected social welfare, expressed as the total utility obtained by all the peers (those without access to the common good have utility zero) minus the total cost incurred by all the peers. It is very important to note that, in the most naïve approach, there is no guarantee that the welfare experienced by any single peer would be positive, that the peers will actually reveal their true utility functions (rather than bogus ones that decrease their costs), or that the total cost needed in the mechanism could be actually supplied by participating peers (the mechanism might not be *budget balanced*). Thus, further constraints must be introduced: a *feasibility constraint* that caters for budget balancing, a *individual rationality constraint* that caters for positive welfare for each single peer as a precondition for participation in the system, and a *incentive-compatibility* constraint that ensures that peers will have higher welfare if they truthfully report their utilities. Antoniadis et al. approach the optimisation problem analytically, using Lagrange multipliers. Although the complete solution of the problem is not presented, an approximate one when the number of peers $n \to \infty$ is discussed. In this case, a single benefit multiplier θ is found below which no peer participates in the peer-to-peer overlay. This implies that every peer that does participate in the overlay pays a flat fee $f = \theta u(Q)$. The authors then apply the model to peer-to-peer incentives for file sharing, but keep the discussion at a strictly theoretical level. There is no mention as to how payments could be implemented in a decentralised fashion.

2.5.5 Systems based on Auctions

One of the problems that currency-based systems have is the pricing of peer resources. A simple way to achieve this with no central control, and if the peers can calculate their own valuations for resources, is through *auctions*. An example of this is CompuP2P [154], a system that implements an open market for peer resources quantised into different markets. Each market is managed by a particular peer through a Chord [231] DHT. Pricing is arrived at by using iterated single-item, sealed-bid, second price auctions (known as Vickrey auctions [306]) as an information-revelation mechanism, and the system is designed to be robust in the presence of self-interested decisions of resource providers and users.

Another auction-based system is Spawn [309], where a distributed CPU resource allocation problem is solved by using an open market. Money in this virtual economy becomes an abstract form of priority, so that better funded processes can obtain correspondingly better access to the computing infrastructure than others. The system also uses iterated Vickrey auctions, and each of the economic actors taking part in the economy for CPU cycles maintains a *resource manager* that manages auctions and assigns CPU time slices accordingly.

The previous two examples of auction-based systems used only single-item auctions. This is representative of much of the work in the literature - few studies use full combinatorial auctions, since they can be very computationally intensive and involve large delays [91]. Sometimes, simplifications are made to make combinatorial auctions tractable. In [119], each participant allocates its finite budget to bid on a given resource set, and receives a proportion of each resource commensurate with the proportion of its bid with the bids of other participants; this same technique was later on proposed by [201] as a replacement for the unchoking policy of BitTorrent (see Section 2.6.1). The Nash equilibria of this game are analysed, and it is unsurprisingly found that they depend on the precise structure of the utility functions for the participants. However, for practical simulation cases, it is found that the Nash equilibria attained have efficiencies close to the social optimum. Additionally, the equilibria are shown to exhibit good *utility uniformity* (the difference between the minimum and the maximum utilities experienced among the users is low) and *envy freeness* [305] (the utility that a participant enjoys given its outcome is close

to the utility that it could have enjoyed had he received the outcome of any other participant).

Another auction-based resource management implementation is Mirage [79], where combinatorial auctions using a centralised virtual currency environment are used for sensor network testbed resource allocation.

Some peer-to-peer mechanisms use auction-based techniques, even without explicitly stating so. One example of this is GNUNet [149], that uses iterated single-item, sealed-bid, first price auctions [227] to resolve resource contention (only the incentives aspects of the protocol are detailed in [149]; see [47] for the anonymity related protocol aspects).

Essentially, GNUNet implements a trust-based economy, where each peer maintains a trust rating for every peer it interacts with. If a peer takes the role of a client, it is making use of overlay network resources and is expected to *lose* trust in the eyes of other peers, whereas a peer taking the role of a server is donating its resources to the peer-to-peer system and is expected to *gain* trust in the eyes of other peers. When a peer requests a protocol action that implies a cost (such as forwarding a given message), it associates with the request a number (called the *priority*) that measures the importance that the sender places in its request being satisfied. The priority becomes, essentially, its bid.

If the priority of the request is high enough to be allocated service, the peer servicing the request then decreases its trust on the requester by subtracting the priority associated with the request (if a peer is insufficiently trusted by another peer to carry this subtraction out, its request is treated as having zero priority). When the requester receives the response, it will increase its trust in the responder by adding to it the priority of the related request. In this way, a *payment* has been effected from the requester to the responder by only modifying local accounts. Therefore, each peer's trust is not under its control: it is under the control of its peers. Additionally, given that trust itself encodes information regarding the best peers with which to carry out transactions, it is to the best advantage of every peer not to arbitrarily modify their trust in other peers (from the point of view of currency, this means that the wealth of other peers is under its control - we expand more on this idea in Section 4.6, where we detail formal mechanisms against this problem). In GNUNet trust is not used to store value, nor is it used to limit access to resources. It is used to **prioritise** in case of excessive demand. In case of resource contention, higher priority requests are served preferentially over lower priority requests, as long as their originating peers are sufficiently trusted to support the corresponding priority. By allowing more trustworthy peers to set higher priorities, the protocol rewards peers that have served urgent or important requests, and in so doing, increased the utility of the peer-to-peer system.

In effect, the priority mechanism of GNUNet implements an iterated single-item, sealed-bid, first price auction where peers bid for service timeslots and pay using trust. In [149], Grothoff repeatedly states that GNUNet is not based on currency, since each peer can arbitrarily increase or decrease the trust it places in other peers, thus allowing wealth to be created (the responder can choose not to decrease its trust on the requester, while the requester does increase its trust on the responder). However, if one envisions a trust-based economy where currency represents trust and each peer is capable of printing its own money as long as it can uphold the promises embodied in it (such as the WAT system [173, 273]), it is clear that GNUNet is in fact a currency-based market that is able to transfer idle network resources into monetary value (currency), and those peers that are able to command more resources for the benefit of other peers will become wealthier. Additionally, the system is self-policing given the rationality of the participating peers: peers cannot "steal" money, and destroying other peers' money is not in their best interest as it destroys information that can be used to increase its own utility by preferring peers that have provided better levels of service. GNUNet solves the "first interaction cooperate" element of the iterated *Prisoner's Dilemma* by allowing untrusted peers to send requests with zero priority and be served by idle peers. Therefore, newcomers to the network can use surplus resources while they build

up a reputation and become more competitive in the resource priority auctions. Unfortunately, GNUNet is susceptible to collusion and sybil attacks.

2.5.6 Systems implementing demurrage

Usually, the use of currency in peer-to-peer systems has been based on a central bank issuing digital cash and verifying transactions to prevent multiple spending of the same "digital coin". However, there are at least two different currents of research that enable single peers to operate their own currencies without the need for a central banking infrastructure. One is based on nested digital signatures [274, 321, 302, 125, 273], while the other is based on small microcredit accounts that are periodically reconciled [159, 248, 149].

Independently of the details of their implementation, markets based on virtual currencies offer the unique possibility of trying out economic policies that would be difficult to implement explicitly in the "real" economy. One of these is the implementation of money with *demurrage*.

In its simplest sense, demurrage is a cost imposed on the accumulation of currency, which could be interpreted as a negative interest rate (See Appendix A.2.6.1) or as hyperinflation. The main reason why demurrage has been explored in the context of network resource allocation is that it provides a negative incentive on the accumulation of wealth, which in turn makes currency closer in its characteristics to network resources that are impossible to store (such as CPU cycles, bandwidth or transmission spectrum).

We now present some attempts to use demurrage in the peer-to-peer system research literature. Similarly to [294], in [171] Irwin et al. propose a scheme in which currency expires a fixed time after being spent, making hoarding difficult. To avoid currency depletion and starvation, credits are automatically recharged periodically. By using a central bank to process auctions, the system can implement flexible recharge time policies.

In [274], Saito et al. present a WAT system with demurrage as a basis for the exchange of *atoms*, *bits* and *presences* - the three main kinds of digital goods, according to the authors. *Atoms* can be stored, but are difficult to duplicate. *Bits* can be stored, but they can be reproduced at negligible cost, and *presences* cannot be stored nor reproduced (they are akin to labour). Saito et al. propose a system in which *bit* producers will give the content away for free, but will pay with special currency that experiences rapid demurrage. In this way, all the other participants that forward these WAT tickets will share the *bit* producer's debt, allowing them to participate in the economy even while obtaining no direct remuneration from their efforts. By compensating the demurrage rates of the peers in each category, a equilibrium state can be reached in which all peers experience high welfare levels. This is further generalised in [275] as the basis for local production and consumption architectures for peer-to-peer networks.

2.6 Incentives and Resource Allocation - Direct Reciprocity

We define a protocol as based on *direct reciprocity* if the interaction between two peers is only influenced by the interactions between them in the past, and not the interaction between them and other third parties.

Thus, direct-reciprocity schemes are appropriate for long-lived relationships where there is ample opportunity for each of the peers to reciprocate appropriately to the behaviour of other peers. However, it has been shown that direct reciprocity schemes for peer-to-peer networks can converge too slowly to be of practical use [198] (see Appendix A.4).

In [219], Marti and Garcia-Molina use a file sharing peer-to-peer system (very similar to Gnutella) to demonstrate a direct-reputation system in which no secondhand information is used by the peers in creating their reputation tables. Peers use self-signed certificates to identify themselves, and the purpose of the reputation system is to prevent the distribution of non-authentic content in response to valid queries. Marti and Garcia-Molina show that, even without taking into account secondhand information,

a reputation system vastly increases the robustness of a peer-to-peer system against malicious users. In this case, the reputation system is acting a sanctioning device to enforce a social rule in which peers that feed bogus content are shunned from further interaction.

The most widely deployed system using direct reciprocity as an incentives and resource allocation mechanism is BitTorrent [84, 20, 21], and it has been repeatedly analysed by the research community. We now discuss a representative sample of this research.

2.6.1 Incentives in BitTorrent

In [84], Cohen describes BitTorrent as a system seeking to achieve Pareto efficiency[10] by allowing peers to find and realise Pareto improving interactions through a *Tit-for-Tat* strategy. Several aspects of the protocol are detailed, including peer bootstrapping, fragment and sub-fragment management and pipelining, fragment dissemination techniques (including the *rarest first* policy) and choking algorithms (including the *optimistic unchoking* and *anti-snubbing* techniques).

One of the best-known elements of the BitTorrent protocol is its support for *Tit-for-Tat*, which has been studied from many viewpoints, particularly its tendency to assign loads to peers in an unfair fashion [298, 49, 254]). For instance, in [254] Piatek et al. argue that the incentive structure of BitTorrent is not robust to selfish clients and that "high capacity peers provide low capacity peers with an unfair share of aggregate swarm resources". In general, Piatek et al. argue that all BitTorrent peers make a large number of altruistic contributions: low bandwidth peers almost never upload enough to others to compete with other peers, and consequently almost never attain reciprocation, while high-bandwidth peers upload much faster than the strict minimum required to attain reciprocation[11]. This can be purposefully avoided by strategic peers to minimise their costs and increase their utility.

In addition, even though low-bandwidth peers almost universally fail to attain *Tit-for-Tat* reciprocation in BitTorrent, they still experience downloads that are significantly faster than their uploads. This is because they receive the benefit of arbitrary optimistic unchoking slots, both from seeders and leechers.

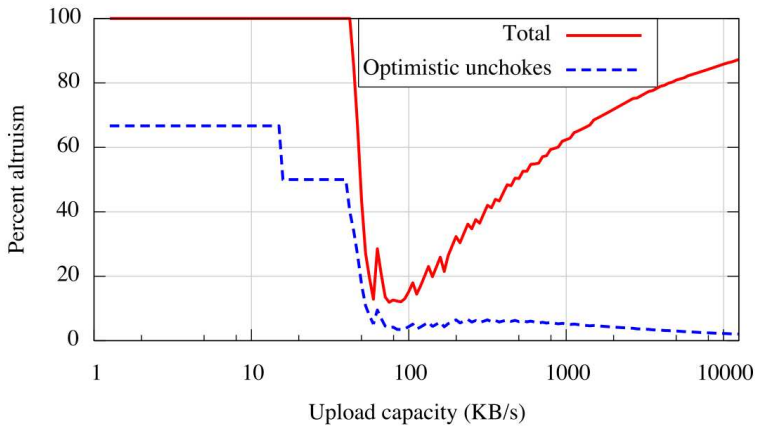


Figure 2.1: Expected percentage of upload capacity not resulting in direct reciprocation (figure taken from [254]).

Thus, as seen in Figure 2.1, high-bandwidth peers tend to give most of their upload capacity "for free". As a proof-of-concept for these ideas, Piatek et al. propose *BitTyrant*, a BitTorrent implementation where each peer ranks all its neighbours by their upload/download ratios and preferentially unchokes

[10] An allocation of a given set of resources to a given set of economic agents is *Pareto Optimal* or *efficient* if it is impossible to find an alternative allocation that makes at least one agent better off without making any other agent worse off.

[11] Piatek et al. find that after a given level of upload (in the region of 10 to 100 kBps), reciprocation is virtually assured. All contributions above this level do not yield in increased reciprocation, and are therefore altruistic.

those peers with high ratios, thus rewarding back cooperation in a reciprocative fashion. Additionally, *BitTyrant* dynamically increases its upload speed to selectively provide an incentive to other peers to maximise their upload throughput (it is usually not necessary to upload at full speed to get full speed downloads from other peers). Under these conditions and an experimental setup including many seeder peers, substantial gains are reported. Other relevant engineering objectives behind *BitTyrant* include:

- **Maximising reciprocation bandwidth per connection**. BitTorrent leechers (non-altruistic peers) reciprocate equally among all the peers they are uploading to. It follows that the upload capacity of a mainline BitTorrent peer is equally partitioned among its reciprocating peers. *BitTyrant* attempts to find reciprocating peers for which the total download bandwidth obtained through reciprocation is the largest.

- **Maximising the number of reciprocating peers**. Providing an incentive to reciprocate to any given peer entails a *cost* in BitTorrent, in the form of lost upload bandwidth. Therefore, instead of just having a hard-coded number of peers to interact with, BitTyrant peers attempt to increase the number of reciprocating peers in their "active set" until the marginal benefit of having one extra peer balances out the extra bandwidth cost imposed by eliciting reciprocation.

- **Deviating from the "equal split" policy**. The benefit that a *BitTyrant* peer obtains on a *per-connection* basis is maximised by reducing its upload to the minimum value required to elicit reciprocation from each of its current peers. Thus, the per-connection investment is minimised, the number of connections is maximised and, as each connection is usually with a non-strategic peer (an altruistic implementation, such as *Azureus*[3], *BitComet*[4] or *μTorrent*[14]), the total amount of download bandwidth is maximised.

Piatek et al. provide evidence that *BitTyrant* performs well in an environment dominated by altruistic implementations, not only increasing performance but making it more predictable and robust. In an environment built on PlanetLab consisting only of *BitTyrant* peers, altruism and performance both *increased*. This is due to the fact that, following [149] and others, *BitTyrant does not withhold bandwidth that would otherwise be idle*. This means that, in this case, strategic peers allocate swarm resources better, but without becoming a hindrance by refusing altruistic service if it incurs close to zero marginal cost.

However, Piatek et al. discuss the following problems with *BitTyrant*:

- *BitTyrant* attempts to contribute as little as possible to elicit reciprocation, but will not withhold unused, available upload bandwidth if it requested. Thus, if the minimum upload to attain the maximum download that an strategic peer can extract from a given swarm does not exhaust its upload bandwidth, the peer will donate the rest of it altruistically - unless it can devote it to get higher download speeds on *another* torrent. This means that, if high-capacity peers seek to maximise the number of torrents they participate in, their upload capacity will be diffused across every one of them, and the average performance for each individual swarm (and the majority of peers) will degrade. This is specially true for low-bandwidth peers that are unable to participate in more than one swarm successfully, and must then settle for the bare minimum performance available. Therefore, even though the total performance of the system across all swarms increases, the individual performance of each swarm degrades.

- New users experience a lengthy bootstrapping period where their download capability is severely degraded. This is because, when a peer joins a swarm, it is initially severely constrained in the number and variety of fragments that it can trade. Thus, even if it has appropriate upload capacity,

it will be unable to amass enough fragments to maintain high upload speeds long enough to elicit reciprocation.

- *BitTyrant* will constantly reduce its upload rates, in an attempt to find the minimum upload bandwidth necessary for reciprocation. Thus, even if there are many high-capacity peers, *BitTyrant* peers will end up with large numbers of low-bandwidth peers, which might impact performance due to deleterious TCP interaction effects [230].

Another example of work regarding the strategic exploitation of BitTorrent is [207], where Liogkas et al. analyse three additional techniques that selfish peers can use to obtain increased performance from BitTorrent:

- **Downloading Only From Seeds**. A downloader repeatedly contacts the tracker in an effort to obtain the IP addresses of all seeds, to contact them all and download from all of them simultaneously. If there are enough seeds, the strategic peer can sustain high download rates only by leveraging the unchoke windows of a constantly varying subset of seeds[12].

- **Not honouring optimistic unchoking**, where the strategic peer infers the download capacity of other peers by monitoring their "new fragment available messages", and from it their upload capacity. Then, the peer interacts only with the fastest uploaders using *Tit-for-Tat*.

- **Advertising false pieces**, where the strategic peer advertises that it has a fragment that it really does not have, so that when another node requests it it can obtain a number of fragments or subfragments and reciprocate with random data.

Liogkas et al. report limited improvements when downloading only from seeds, but their results are inconclusive, as their experimental setup included only one seed. As expected from our discussions in transaction costs (see Appendix A.2.1), they find that not honouring optimistic unchoking is counterproductive because it is very hard to actually find the best peers without it. What this tells us is that optimistic unchoking, even with its exploitation risks, remains critical in assessing peer QoS and finding better peers. The indirect estimation algorithm that appeared to have a limited success in PlanetLab [16] fails in the wide area Internet, and Liogkas et al. indirectly provide evidence to suggest that information gathering (network sensing; QoS estimation/mapping) is a good incentive to share upload bandwidth through optimistic unchoking. Finally, advertising false pieces has limited effect as well, as the clients themselves (as opposed to the BitTorrent protocol) implement a direct reciprocity system where, if a given peer has sent invalid fragments, it is shunned from further interaction (*banned*). Thus, even though they state that the endogenous mechanisms of BitTorrent are robust against strategic agents, their experiments were made in a very specific circumstances and their results might not be applicable to other environments.

Another example of an analysis of the strategic scope for BitTorrent clients is [209], where Locher et al. discuss a technique to download entire torrents without any kind of reciprocation. Locher et al. provide a software implementation: *BitThief*, that specifically exploits optimistic unchokes - the main peer performance exploration tool of BitTorrent. A *BitThief* client does not upload at any time, and thus does not perform any chokes or unchokes for remote peers. Additionally, it never announces availability of any pieces: all other peers can only assume that the *BitThief* client is a new peer that recently joined the swarm. In order to sustain a good download rate, *BitThief* relies heavily on optimistic unchokes by the peers to which it connects. As the *BitThief* client will not reciprocate, the download bandwidth

[12]This has been explored repeatedly, in particular as part of [209] and [286], where the technique of connecting to a much larger peer set than the one prescribed by the BitTorrent protocol specification is in this case called the *Large View Exploit*.

obtained per peer in this manner is severely limited. To obtain a high download speed, then, *BitThief* needs to connect to a very large set of peers. As optimistic unchoking slots are effectively uncorrelated between agents, download bandwidth for *BitThief* clients is linearly dependent on the number of peers in their active uploading set. Therefore, instead of limiting themselves to 80 connections (as the "official mainline" client and others do), *BitThief* clients repeatedly contact the tracker to obtain as many leecher and seeder IP addresses as possible, and connect to all of them (Locher et al. state that for some torrents the number of active TCP connections per BitThief client grew to more than 500[13]). Additionally, the *BitThief* protocol refuses to follow the *rarest first* fragment download policy. Instead, *BitThief* clients maximise their download speed by downloading whatever fragments they need that are available in high-throughput peers, putting their own download speed before system-wide concerns like adequate fragment diffusion[14]. In general, Locher et al. find that for swarms with many peers (in particular seeders), and torrents for small files, *BitThief* outperforms the "official mainline" BitTorrent client.

BitThief can exploit BitTorrent communities that enforce upload/download ratios by purposefully lying to the tracker, reporting arbitrarily high amounts of upload traffic in order to maintain a proper ratio while freeloading. This attack can be compounded with a *sybil attack*, reporting the existence of arbitrarily many peers.

Other exploits that Locher et al. tried which were more successfully countered by BitTorrent are detailed below:

1. **Announcing false pieces**. In [207], Liogkas et al. propose that BitTorrent can be exploited by strategic peers if they advertise fragments that it they do not have yet, alongside with fragments that they actually have. When answering a request, the strategic peer uploads either random data (if it has not got the appropriate fragment) or the fragment itself. Doing this for complete fragments is harmful for the strategic peer, as many BitTorrent client implementations (in particular, the "official" client and Azureus) will ban peers for fragment hash checks that fail. Thus *BitThief* will answer all fragment requests with random data, but will avoid uploading a complete bogus fragment - it will purposely fail to upload the its last sub-piece. When this sub-piece is eventually downloaded from a different peer, its CRC will fail and this peer will share the blame for the bogus piece, as the affected peer will be unable to ascertain which of the two sent the offending sub-piece or sub-pieces, and will be unable to retaliate effectively[15]. Though the "official mainline" client can be exploited this way, Azureus is resistant: it will request the whole fragment again, but will download it *only* from the peer that uploaded the most pieces when the hash check failed. If the peer fails to re-deliver the fragment or refuses to upload and the connection stalls, the peer is banned. In general, the authors find that, due to adequate protection measures in client implementations, uploading random data is not helpful in any way to improve download speed.

2. **Sybil attacks against given peers**. *BitThief* benefits from the optimistic unchoke slots of other peers, and therefore it would be advantageous for *BitThief* clients to be counted several times in each client. This, however, would entail setting up parallel TCP connections to the same peer, and most BitTorrent implementations guard against this by immediately shutting down any TCP connection attempts from any IP address with which there is an active connection. Of course, multihomed peers (or peers with access to a series of anonimising proxies, such as *Tor* [102]) would be able to execute this attack successfully.

[13]The authors state that, in their experience, there is no performance penalty with the simultaneous opening of multiple TCP connections as reported in [230].

[14]It may be said that the rarest first policy is also rational for selfish peers, as it maximises the probability of successful *Tit-for-Tat* transactions later on for the downloading peer. However, as *BitThief* does not reciprocate at all, this does not apply to it.

[15]This corresponds to a *hidden action* situation (see Chapter 6).

2.6.2 BitTorrent Communities and Multi-Torrent Collaboration

It is well known that animal communities can facilitate cooperative behaviour by providing and regulating social capital (See Appendix A). This same approach is valid in the context of peer-to-peer protocols; one example is [33], where Andrade et al. analyse the effect of BitTorrent communities on users' cooperative behaviour. One of the interesting results in [33] is that in torrents with many leechers and few seeders, BitTorrent is successful in penalising freeloading, as shown by freeloaders experiencing increased download times for a given file. However, for torrents with large numbers of seeders (high seeding ratios), freeloaders may achieve download speeds comparable (or higher) to those of contributing peers. The authors posit that it is the competition between the uploads and the download TCP acknowledgements that impacts the peer download performance.

Andrade et al. further analyse the seeding behaviour of peers in a particular community site (*etree* [8]), finding that the *seeding ratio*[16] tends to be higher for torrents of smaller size and for younger torrents. They find that torrents with a high *sharing ratio*[17] tend to have high seeding ratios and large numbers of participating peers, and that those community sites that imposed sharing-ratio enforcement tend to have higher levels of seeding. In this case, seeding is not altruistically motivated: it stems from an external incentives structure built around the community site and the easy access to content that it offers. In essence, the community site gives peers the benefits usually associated with social capital: information diffusion, access to privileged knowledge, and increased trust in transaction outcomes. Sites impose their own set of "social rules", and noncompliance is punished with denial of access to social capital (community resources). Andrade et al. show that this risk of losing social capital is enough to change the incentive structure of BitTorrent users, which then change their behaviour to a more "reciprocally altruistic" one.

In [152], Guo et al. focus on multi-torrent collaboration. According to their research, even though most BitTorrent studies have focused on single torrents, around 85% of all peers participate in multiple torrents. The paper starts analysing single-torrent statistics, with the following conclusions:

- **Peer arrival rate to a given swarm decreases exponentially**. Thus, given that seeds remain in the system a relatively small time, fragment availability in BitTorrent degrades very quickly, so that completing a full torrent download is only feasible in a small time window after its publishing. This is only a reflection of the consumption patterns of BitTorrent users: file availability in BitTorrent is heavily dependent on a large number of leechers being present (what has usually been called *swarm health*), and this is in turn dependent on torrent popularity and the usage dynamics of torrent distribution sites.

- **BitTorrent swarm performance varies widely from swarm to swarm**. Peers in swarms with large populations have higher (and much more stable) download speeds when compared with those in small swarms. For these, download speed is much lower, has much greater variance, and is easily affected by the individual behaviour of seeds.

- **Current BitTorrent implementations are unfair to high-capacity peers**. Guo et al. find that upload/download ratio *decreases* with increasing downloading speed. Thus, as peers extract higher download bandwidths from the system, they have to invest less of their own upload bandwidth. One possible explanation for this, presented in [152], is that faster peers finish downloading the torrent content faster, and then immediately leave the system. This counters the common notion that peers remain active for approximately the same amount of time in a given torrent (say,

[16] The seeding ratio is defined as $\frac{n_u}{n_u+n_d}$, where n_u is the number of uploaders and n_d is the number of downloaders.

[17] Andrade et al. define the sharing ratio of a peer as the quotient between the total amount of data the peer has uploaded and the total amount it has downloaded. This can be extended to define the sharing ratio of an entire torrent to be the total amount of data uploaded by all its active peers divided by the total amount of data downloaded by all its active peers.

overnight), but peers with better download speed finish more quickly and thus spend more time as seeders.

Guo et al. then move on to propose a model for inter-torrent collaboration for BitTorrent. The model that they propose is essentially analytic, although some aspects of protocol design are addressed. In particular, Guo et al. suggest forming an overlay network among BitTorrent trackers (the *tracker site overlay*). Thus, when a peer wishes to join a swarm, it informs the tracker of this swarm of its available files. The tracker then contacts the trackers responsible for swarms that include peers with whom the original peer can trade, and peers are then introduced with each other. Through the use of the tracker site overlay, it becomes possible for peers to contribute bandwidth in a given torrent while downloading from another, which enables indirect reciprocity.

In [256], Pouwelse et al. present a measurement study on BitTorrent and and associated search website (in this case, *suprnova.org*). Pouwelse et al. find that system availability for BitTorrent networks is very sensitive to the availability of their centralised components, such as trackers and search websites. With respect to the actual peers, Pouwelse et al. find that only 17% of the peers remain seeding longer than an hour after their download finished. This reduces to 3.1% for 10 hours and 0.34% for 100 hours, underscoring the fact that long-lived peers on which central system functions can be delegated can be difficult to find[18]. This behaviour, however seems to be dependent on the content. In [290], Stutzbach and Rejaie analyse the churn behaviour of peers in Linux distribution torrents, finding that their results are heavily influenced by a large number of stable infrastructure seeders (this is in addition to the increased altruism reported in [172] for Linux Red Hat 9). In the words of the authors:

> *At any point of time, a majority of participating peers in the system are long-lived peers. However, the remaining small portion of short-lived peers join and leave the system at such a high rate that they constitute a relatively large portion of sessions.*
>
> (quote from [290])

It follows that, although sufficiently high levels of peer availability might be attainable in a pure peer-to-peer architecture, resilient designs for stream distribution and peer search should adequately replicate functionality among long-lived peers, and should also provide an adequate incentives structure that helps increase the average peer uptime. Pouwelse et al. correctly point out that BitTorrent does not provide an incentive for seeding behaviour, as it requires altruism on the part of the user, and posit that allowing seeders more expressive control over their altruistically donated bandwidth utilisation might be beneficial in this regard.

2.6.3 BitTorrent Locality Awareness

Most studies focusing in BitTorrent assume that, when peers select neighbours to barter with, they do so in a random fashion[19]. This leads each peer to generate a large volume of potentially long-distance, inter-AS TCP traffic. The reduction of this inter-AS traffic has generated increasing interest in the research community, with the objective of developing peer selection procedures that could maintain or increase the performance of BitTorrent while, at the same time, generating topologies engineered to be of higher performance than a random graph graph with regards to to specific metrics. One such attempt is [50], where Bindal et al. examine a peer selection technique for BitTorrent (*biased neighbour selection*)

[18] It may be argued that if the peer lifetime distribution is heavy-tailed, there will always be a non-negligible probability of finding a peer with arbitrarily long lifetime. This, however, does not mean that this peer would be able to cope with the load of reliably managing an overlay network of Internet scale. Additionally, even though the probabilities are non-negligible, they are certainly extremely below the availability requirements of any Internet application.

[19] Or, rather, the trackers select randomly for them.

in which each peer attempts to select the majority of its neighbouring peers from those within the same ISP.

Using simulations, Bindal et al. show that the performance of BitTorrent can be maintained while reducing significantly the inter-ISP traffic. However, this requires that the original seeder has high upload bandwidth (in the order of four times the current average bandwidth per peer). More generally, Bindal et al. state that any admissible peer selection algorithm aiming to optimise any topological or traffic engineering objective can give good results, provided that the seed has large bandwidth and that a small amount of randomness is maintained during peer selection.

When using locality optimising algorithms such as the one proposed in [50], the variance of the file download time distribution decreases. This is because the real availability of different file fragments is better estimated in this case than in normal BitTorrent; as the resulting topologies will be much more clustered within ISPs, different peers will have much more similar fragment availability estimates, and will therefore replicate pieces in a more locally optimum way.

The most important reason that BitTorrent maintains its performance when supplemented with locality-aware trackers is its *rarest-first* fragment replication policy. Peers within the same cluster have a very clear view of which fragments are inside it. Furthermore, once a new fragment has been "imported" into the cluster, it can very quickly reach many internal peers, thus reducing the probability of a single fragment being "imported" several times into the cluster. As a result, most of the fragments that traverse cluster boundaries tend to do so a only a very small number of times, increasing the efficiency of the system.

2.6.4 Analytic Models of BitTorrent

Although analytic approaches might have limited applicability to practical scenarios, they are of great importance because of the insight that they yield into the essential issues of the problem under study. In [221], Massoulié and Vojnović propose an analytic model for fragment-based, swarming peer-to-peer systems. This model, although requiring some basic assumptions for closed-form solubility, is general enough to encompass many kinds of swarm-based distribution systems. Massoulié and Vojnović model fragment diffusion using a *density dependent Markov process*, proposing a system of differential equations to explore their dynamical properties, such as equilibria and convergence basins.

In [258], Qiu and Srikant propose a fluid flow model of BitTorrent and compare its performance with measurement traces. This model is based on a number of simple assumptions, such as exponentially distributed leecher and seeder departures. Special attention is given to the analysis of the equilibrium points of the resulting dynamical system. By considering in turn the equilibrium points of the system when dominated by its upload or its download bandwidth, they derive a closed form solution for the equilibrium point and examine it under different parameter values, showing that the analytic model yields sensible results that confirm the intuitions that one might get from the protocol itself.

The stability of the equilibrium point is then analysed by linearisation, and found to be stable. Qiu and Srikant further generalise the model to encompass Wiener processes, and propose expressions for the expected variances of the number of seeders and leechers. A game-theoretic model of BitTorrent is then proposed, and then used to explore the existence of Nash equilibria as a function of peer physical bandwidth. In the case of identical bandwidths with rational, omniscient peers, they prove that no Nash equilibrium exists. In a similar case, but where peers form groups whose members have identical upload bandwidths, they prove the existence of a Nash equilibrium where each peer chooses to saturate its physical upload bandwidth. Additionally, they explore optimistic unchoking and, show that a freeloader will be able to obtain, on average and with no optimisations (as compared to *BitTyrant* [254] or *BitThief* [209]), around 20% of the maximum possible downloading rate. They compare their model with simulations and show that, in the mean, the fit is good.

Finally, Qiu and Srikant create a BitTorrent swarm and measure its number of seeders and leechers, and compare this with the results from their model. The variations from the fluid flow model, once it reaches steady state, are consistent with the 95% confidence intervals to be expected from the variance of the stochastic process.

2.6.5 BitTorrent Measurement Studies

The characterisation and measurement of the characteristics of BitTorrent-based systems under load has received attention from the research community. In [172], Izal et al. follow the Linux RedHat 9 torrent, both from the tracker and the peer perspectives. They report that, on average, peers remain for 6.5 hours as seeders after finishing the torrent download[20]. Additionally, Izal et al. report that seeders had a very important effect in the distribution of this torrent: seeders contributed more than twice the amount of upload bandwidth that leechers contributed, while the seeder/leecher proportion was consistently higher than 20%[21]. The average download rate in [172] is consistently high, above 500 kbps. However, peer bandwidth exhibits high variability, spanning more than three orders of magnitude ($\sim$ 10kBps to $\sim$ 10000kBps).

An important characteristic of BitTorrent is that it scales well with flashcrowds: when new peers arrive at a high rate, the download bandwidth of the system is not evenly divided between all peers (which could lead to stalling, and no peers finishing their downloads to become altruistic seeders that increase overall bandwidth availability). Instead, download bandwidth is preferentially allocated to older peers, because they have a wider selection of fragments to upload and therefore can more easily trade for the fragments that they still need. Thus, the probability of having an idle download due to lack of fragments to reciprocate with is much greater for newcomers, and the probability of *some* peers finishing the download to become seeds increases. However, this same effect works against new peers attempting to bootstrap their participation in the swarm. In this case, however, the *rarest fragment first* policy facilitates the inclusion of new peers, as these will attempt to download (initially through optimistic unchokes) those fragments that will be most interesting to the swarm, increasing their own upload capability. In this case, a peer-level selfish incentive (maximise the trade capability of the fragments downloaded) becomes instrumental in generating a system-wide architectural design: maximise the probability that all fragments can be easily and quickly downloaded from any peer the swarm. This simplifies protocol design, as it decouples fragment search from topology awareness, increasing system scalability and swarm health.

In [205], Lien analyses some problems with the BitTorrent protocol on low-bandwidth, asymmetric wireless links. In particular, Lien very briefly explores different topologies for fragment distribution trees in an attempt to balance the upload bandwidth for each peer and the delay that the fragments experience from their source. Lien proposes solving the problem as a distributed *Minimum Spanning Tree* with a restriction on the maximum number of adjacent nodes.

An important issue that is garnering attention in the research community is the study of the relationship between overlay networks and their underlying Internet. In [170], Iosup et al. study the relation between the IP substrate and the application layer BitTorrent overlay, correlating some of its measured characteristics with those of its underlying Internet. To do so, Iosup et al. inserted monitoring probes in the top 2000 BitTorrent swarms advertised on *thepiratebay.org* [15]. One of their results is that Europe is the dominant continent for BitTorrent, both in the number of users it has, and the amount of data

[20] This might seem overly high in the light of the results in [256], but it is easily explainable in terms of differing user incentives. In the case of [256], the content corresponds to a software product that could be being distributed in violation of piracy laws, imposing a liability risk on users. Red Hat 9, on the other hand, was a perfectly legal download that entailed no added risk.

[21] Again, these data seem overly high for a typical torrent. User incentives might play a role in proposing an answer to this as well: In general, linux distribution enthusiasts derive psychological utility from widely distributing their favourite distributions, thus helping them acquire more users. Moreover, it might be argued that Red Hat 9 users, as part of the open-source user community, have a more altruistic ethos (or a higher "warm-glow" utility) than an average Internet user.

transferred per user: 59.4% of BitTorrent users are located in Europe, and they account for 58.8% of all transferred traffic. North America follows, with 22.3% of the users. The USA accounts for 14.4% of all recorded BitTorrent users (compare with 7.7% for the UK)[22]. Some results presented by Iosup et al. are reproduced here in table form:

Connection	Direct	Strong	Good	Average	Loose	Very Loose
Number of IP Path Hops	0-1	2-4	5-9	10-14	15-19	20+
Percentage of all Paths	5.9	20.4	40.2	25.3	6.4	1.1

Table 2.1: Distribution of IP path hops for PlanetLab BitTorrent probes and the peers which contacted them. Table reproduced from [170].

Connection	Direct	Strong	Good	Average	Loose	Very Loose
Number of AS Traversals	0-1	2	3	4-5	6-9	10+
Percentage of all Paths	12.5	44.7	32.0	10.2	0.5	0.1

Table 2.2: Distribution of AS traversal hops between PlanetLab BitTorrent probes and the peers which contacted them. The authors found a number of *traversal loops*: AS traversals in which the same traversal occurs more than once. They report that at least .01% of all the analysed traceroutes exhibit this peculiarity, leading to as many as 56 AS traversals in a single IP path. Table reproduced from [170].

Connection	Direct	Strong	Good	Average	Loose	Very Loose
Hops within each AS	1	2	3	4-5	6-9	10+
Percentage of all Paths	58.1	12.9	8.0	12.2	7.6	1.2

Table 2.3: Distribution of IP path hops within single ASs between PlanetLab BitTorrent probes and the peers which contacted them. The authors found a number of intra-AS *routing loops*: IP paths on which the same IP address occurs more than once. They report that at least .001% of all the analysed traceroutes exhibit this peculiarity. Table reproduced from [170].

	Local	LAN	Campus	Metropolitan				Inter-City		Longer
Min. RTT (ms)	0	15	50	100	200	300	400	500	750	1000
Max. RTT (ms)	15	50	100	200	300	400	500	750	1000	+
% of all Paths	0.6	2.9	5.8	13.3	13.7	13.8	8.6	12.8	7.5	20.7
Total	0.6	2.9	5.8	49.4				20.3		20.7

Table 2.4: Distribution of IP traceroute-measured Round-Trip times between PlanetLab BitTorrent probes and the peers which contacted them. The authors only report results for successful traceroutes. Additionally, as the measurements were taken from points with high bandwidth connections (PlanetLab sites), there might be a systemic bias towards smaller RTTs (the RTTs for arbitrary Internet connection points may be larger). Table reproduced from [170]

2.7 Incentives and Resource Allocation - Indirect Reciprocity

We define a protocol as based on *indirect reciprocity* if the interaction between two peers is influenced not only by the interactions between them in the past, but also by the interaction between them and all other peers. Although the majority of studies relating to indirect reciprocity have focused on *reputation systems*, we start the analysis of these kind of systems by exploring the ways in which BitTorrent can be extended to go beyond *Tit-for-Tat*.

[22]This, however, might be a reflection of the general trend of Europe to become the largest presence in the Internet, as indicated by the number of Internet users[5].

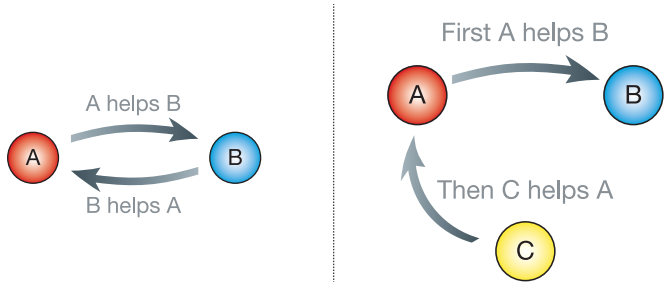


Figure 2.2: Comparison between Direct and Indirect Reciprocity (figure taken from [244]).

2.7.1 Systems based on Social Enforcement

In [255], Pouwelse et al. propose Tribler, a set of BitTorrent extensions that use social relationships between peer-to-peer users to simplify trust management, and clustering using similarity in tastes to aid in content search, discovery and distribution. Their approach involves disallowing anonymous participation on the peer-to-peer network, by binding the BitTorrent client to a personal identity, that is itself embedded in a social network. Downloading performance is improved by allowing trusted peers (those bound to users with strong social ties between them) to use each other as part of a *cooperative download* [130] swarm where a central *collector* peer delegates download operations to other peers, that later forward the obtained fragments for free (without *Tit-for-Tat* or payments). Thus, each peer virtualises itself, at the application layer, into a *peer alliance set* that is, for all non-member peers, a peer with very large upload and download capabilities. The bootstrapping problem is also alleviated, as the peer-to-peer network superimposes itself on the social networks of the users[23]. If no trusted users are directly available at start-up (the system allows communities to form naturally out of tastes or geographic location), a set of *superpeers* are contacted to provide the starting peer with a set of peers (this is based on the *tracker* protocol of BitTorrent). Tribler separates data transfer from peer and content discovery; this latter functions are implemented using a special *overlay swarm*. The social relations, altruism levels, peer uptimes, taste similarity communities and other context information are stored in every peer in *megacaches*, which are later exchanged in form of *Bloom filters*[52] using gossip-based protocols. Additionally, Pouwelse et al. propose a *buddycast* algorithm, that allows each peer to update information with its socially close peers and attempts to discover new peers with the same tastes. These new relationships can be cultivated, and the peers could eventually become socially close.

In [130], Garbacki et al. detail 2Fast, a collaborative downloading strategy that could be considered the precursor of the Tribler function of the same name. As a rationale for the work, Garbacki et al. identify a basic weakness in the *Tit-for-Tat* strategy of BitTorrent: it is unable to preserve information about peer contributions between different torrents. Thus, in the case of leecher-dominated torrents, peers can only benefit from the system inasmuch as they can instantaneously contribute to it, and, for asymmetric Internet connections, this means that download rate will be capped at a level that correlates roughly with the upload capacity of the downloading peer (this does not happen in seeder-dominated torrents, due to their high levels of altruism). To address this problem, 2Fast proposes the formation of trusted peer groups whose total upload capacity is greater than or equal to the download capacity of any given member peer. Thus, when one of the member peers wants to download a given torrent, it becomes a *collector* and the rest of the peers in the group become its *helpers*. Each of the helpers contributes its upload bandwidth to the collector, and uses its download bandwidth to fetch appropriate fragments for the collector. From the standpoint of an external peer, the trusted group looks as a single, multihomed peer

[23] Originally, Tribler required its own social network, but there is no reason by which it could not use already existing ones such as Facebook [9] or Twitter [26]

with large upload capacity. Intra-group contribution balancing is achieved by collector peers performing as helpers to other peers in turn, when needed. Thus, 2Fast is based on *reciprocal altruism*: Collectors recruit helpers by promising that the bandwidth they invest in increasing the collectors upload capacity (and thus boosting its download capacity) will be returned in the future, when the helper-collector relationship is reversed.

Other authors have focused on the game-theoretical aspects of indirect reciprocity. One example is [198], where Lai et al. extend the *Iterated Prisoner's Dilemma* to model the asymmetric nature of peer-to-peer interactions, and implement an evolutionary strategy to refine peer strategies. In particular, a simulation model is developed in which every peer takes the role of either client or server for each round of the game. Clients choose the servers they wish to interact with, and whether they want to *cooperate* or *defect* in this interaction. A round consists of two games for each player: one as a client and one as a server. After a *generation* of r rounds, each strategy is assigned to a proportion of peers in accordance to the average gain that it gave to its users. Thus, high performance strategies tend to proliferate, and low performing strategies tend to die out. Each peer implements a decision function that takes a history of a player's actions, either in direct interaction (*private history*) or obtained through gossip or centralised reputation systems (*shared history*), and decides whether to cooperate or defect with that player. The authors propose the *reciprocative decision function*, that instructs peers to cooperate with the probability

$$\mathbb{P}(\text{any peer cooperates with peer } j) = \min\left(\frac{\text{number of times } j \text{ gave}}{\text{number of times } j \text{ received}}, 1\right).$$

Additionally, Lai et al. compare between *objective* reputation systems, where every peer agrees on the reputation rating for every peer, and *subjective* reputation systems where each peer weighs the reputations reported by other peers with their relative trust ratings. Through simulation, Lai et al. show that imperfect information on the past actions of peers (private history) can lead to inefficiency, and this issue aggravates with system size. Only very small peer-to-peer systems (smaller than 200 peers in their simulations) were able to function at appreciable levels of cooperation using only private history. Lai et al. finally analyse the issue of *whitewashing*, where peers use disposable identities to exploit generosity towards newcomers. However, the solution that Lai et al. propose hinges on newcomers arriving in batches of "good" and "bad" newcomers, which is by no means a given in practice.

2.7.2 Systems based on Reputation Systems

The study of reputations, their value and the way that they can affect the decision-making process of strategic agents has been a favourite topic of study in economics and mathematical sociology (see [216] and Appendix A). Recently, software platforms have been developed that allow peers to collect, distribute, and aggregate service quality information about the past behaviour of other peers [263]. These have been called *reputation systems*.

One of the classic examples of a reputation system for peer-to-peer networks is *EigenTrust* [186, 185], which works as a participation metric that can be used to reward peers which contribute to the overlay while still allowing less active peers to participate in it. Kamvar et al. very briefly show that the use of EigenTrust correlates positively with a set of participation indicators on Gnutella (number of authentic uploads, popularity of files shared, uptime, number of shared files, etc.), and propose to use larger bandwidth shares and bigger overlay-level search query Time-to-Live (TTL) as rewards for participatory peers.

There have been, however, many other reputation system proposals. One of them is NICE [281], in which trust values for a reputation mechanism are calculated in a decentralised manner by inferring multiple trust paths using probabilistic search with *attenuated Bloom filters* [265]. Peers store *cookies* that constitute trust certificates, proofs that the peers have performed services to other peers. To request

a service from peer n_j, peer n_i recursively searches for cookies issued by n_j starting with its trust neighbours (peers that have issued cookies to n_i). This search is directed by using Bloom filters, and eventually converges to a set of transitive-trust paths between n_i and n_j. In the words of Sherwood et al. [281]:

> *Nodes that request resources present their credentials to the resource owner. Each credential is a signed set of certificates which originate at the resource owner. Depending on the set of credentials, the resource owner may choose to conduct a reference search. The trust ultimately computed is a function of both the credentials, and of the references.*
>
> (quote from [281])

The trust paths calculated as a result of the *reference search* are used to compute a trust value that n_j places on n_i, and further, to modify the terms of the transaction.

The use of probability distributions to compactly encode peer preferences in a pseudo-anonymous way has been researched a number of times. For instance, in [60], Buchegger and LeBoudec propose a Bayesian approach for belief representation and learning, so that second-hand information can be used in a slander-resistant fashion to build reputation tables. Peers in this case integrate (*merge*) the information from other peers using either a *linear opinion pool* (an average of the opinions of other peers, where the information reported by each peer is weighted by its own reputation) or a *independent opinion pool* (the product of the opinions of other peers, as if they were independent random variables). Resistance to slander is obtained by ignoring those reports that significantly deviate from the peer's direct experience at the moment of information merging.

Another example is [99], where Despotovic and Aberer estimate the reputations and trustworthiness ratings between any peer pair without having to explore all trust paths by using a *Maximum Likelihood Estimation* approach. Thus, second-hand information is integrated weighted with a probability of it being true - an independent *credibility* rating that is calculated by comparing reports with individual experience (like in [60]).

Given that the basic commodity to be exchanged in peer-to-peer streaming systems is bandwidth, it is advantageous to focus on reputation systems for bandwidth trading. One of these mechanisms, Havelaar [147], relies in multiple aggregation levels for reputation information, and reporting to a stable (non-changing) set of peers (in a similar note to KARMA [307]). Each peer maintains a vector that has as many components as there are peers in the peer-to-peer network. Each peer populates this vector with its own observations, and sends it to a fixed set of *successor* peers. Each of these successors adds the measurement vectors from all its predecessors, and creates a new vector that is combined with its own observations by means of a simple exponential moving average. This averaged vector is propagated to its successors. By exploiting sparseness in the aggregation vectors, communication overheads can be significantly reduced, favourably comparing to a DHT of equivalent size. Slander is prevented through statistical averaging: given that the opinions of a single given node are very quickly dispersed and mixed with the opinions of many others, only large colluding groups can have important network-wide effects in the reputation system. Havelaar allows for indirect reciprocity and deferred usage of contributions, but it is vulnerable to whitewashing and sybil attacks.

There has been great interest in the study of selfish routing (see, for instance [128, 242, 259, 269] and the references therein), and some of this work can be applied to peer-to-peer networks. A very simple but representative example is [51], where Blanc et al. investigate incentives to discourage free-loading in a routing game. This game is a variation of the random matching game in [187], adapted to overlay routing over Chord [231][24]. Time is divided into discrete rounds during which peers are

[24] This formulation is almost identical to the *Contrite Tit-for-Tat* of [39, 38]

randomly matched and each pair plays a *Prisoner's Dilemma*, with the following stateful strategies:

1. If both peers are *innocent*, they both cooperate.
2. If both peers are *guilty*, they both defect.
3. If one peer is *innocent* and the other one is *guilty*, then the *guilty* player cooperates and the *innocent* player defects.

A peer starts the game with the *innocent* label. Deviation from this strategy is policed by the peers themselves, as it triggers a punishment lasting for τ rounds where the offending peer is branded *guilty*, and then punished by all other players (all will instantaneously defect in its presence). After τ rounds, the peer becomes *innocent* once again, provided that it has followed the strategy during the punishment period - cooperating even though it knows every other peer will defect (if the peer deviates in its behaviour during the punishment period, the punishment is restarted). This implies that before a peer can be forgiven, it must suffer τ rounds of the *sucker's payoff*[25] of the *Prisoner's Dilemma* as a punishment for deviation from the social norm.

It has been proved [187] that following these social rules is a *subgame-perfect equilibrium*. However, the routing game of [51] differs from [187] because it is *asymmetric*: if peer n_i requests peer n_j to route a fragment or request, it is not n_i who can ascertain adherence to the social norm, but rather, this is only possible for any peer n_k downstream from n_j in the overlay routing path. Thus, Blanc et al. assume the existence of an omniscient, trusted central authority that is able to perfectly observe and judge the actions of each peer and modify its reputation value accordingly (this is later relaxed to consider a central authority that only detects misbehaving peers with a given probability).

Blanc et al. identify the following issues with their work:

1. The reputation system does not take into account the fact that different peers impose different loads in other peers. Under large peer heterogeneity, the incentive mechanism can fail.
2. The incentive mechanism is not strong enough to make the social norm strategy *dominant*. Thus, a peer might choose not to follow the social norm if other peers are failing to observe it, and this behaviour can become self-sustaining. Colluding or misbehaving peers can thus derail the socially-enforced equilibrium.
3. In a decentralised peer-to-peer system, reputations cannot be centrally managed, and reputation-bearing messages can be modified, dropped, incorrectly routed or created out of false information. Even if peers cannot tamper with their own reputation scores directly, collusion among peers and sybil attacks can be used to artificially modify reputation ratings.

As evidenced by the inclusion of [187], indirect reciprocity can be motivated by social structure. In [153] Gupta and Somani propose the creation of a reputation management system that allows peers to form trusted communities. These trusted communities form and dissolve dynamically, and from the point of view of external observers they are indistinguishable from single high-performance peers. All peers in a trust group are constantly assessing each other, and may opt to evict under-performing peers if enough votes are presented.

The study of reputation systems has developed greatly in the last 10 years, and space constraints mean that many other possible solutions that have been explored by the research community (such as CONFIDANT [59], Free Haven [101], the image scoring model [243] and many others) will not be directly addressed.

[25] The lowest level of reward available in the game.

2.7.3 Systems based on Contracts (Hidden Action)

Most of the work focusing in the strategic elements of QoS on overlay networks assume that clients are able to equate server effort with the experienced QoS. Thus, although the strategic character of QoS is widely recognised, the fact that it might be impossible to measure it directly has not received nearly as much attention. In this context, the *Principal-Agent* model (see Sections 3.4.1 and Appendix A.4) has been useful in the study of sharing risks and externalities between peers communicating over best-effort networks, and that can thus only imperfectly assess their respective behaviours.

There have been several attempts to model contracts in the context of peer-to-peer systems. In [135], Ghosal et al. propose a system where peers agree to a contract when joining the network. This contract specifies the amount of resources that they will be required to contribute to the system in order to gain access to its services. Ghosal et al. cast the optimal contract calculation as a dynamic programming problem that seeks to assign to incoming peers a set of obligatory resource donations that is mutually beneficial for the peer and the network (both their utilities increase). As a particular example of this, in [191] Khorshadi et al. consider a peer-to-peer file sharing system where each download must be offset by uploading it up to N times within a maximum time interval T.

With respect to the modelling of hidden action in networking scenarios, there have been several attempts to use the principal-agent model in routing and peer-to-peer systems. In [120, 118], Feldman et al. propose a system in which routing is analysed using a principal-agent model. The endpoints of the flow (the sender and the receiver) are modelled as the principal, and the forwarding routers as agents. Feldman et al. derive expressions for optimal contracts that can be used to elicit high effort on the part of the agents, while minimising investment by the principal. Furthermore, Feldman et al. find that the usefulness of information regarding outcomes at intermediate stages in the routing process is contingent on the network topology, and that contracts can be implemented both directly (between the principal and each agent) or recursively (between every node and its downstream partner).

In [229], Moore analyses hidden action from the standpoints of market theory and communitarian resource allocation. Based on qualitative arguments, Moore proposes to constrain overlay network topologies to consist of fully connected cliques interconnected through a smaller number of stable, long lived channels. Thus, Moore posits, peers would be forced to have longer lived interactions with a smaller number of peers, making reciprocative strategies more effective.

Implementing a model inspired by the notion of *social capital* [126, 86], [330] Zhao et al. propose to counter hidden action by careful construction of the overlay topology. They assume that only some peers manifest strategic behaviour, and they are randomly placed in a simulated topology following a power-law distribution in the node degrees. Their proposal can be described, essentially, as supplementing the topology with additional links in order to increase the probability of finding paths consisting only of non-strategic peers. Obviously, as the number of strategic peers in each end to end path decreases, the effect of hidden action decreases as well.

2.7.4 Other Indirect Reciprocity Systems

Some reputation systems follow models that were originally developed for the analysis of reputation formation in humans. For instance, in [243], Nowak and Sigmund propose a model for the emergence of cooperative behaviour among strangers. Agents only meet each other once, and thus there is no scope for reciprocal altruism. Instead, agents label other agents with an *image score* that allows other agents to discover the type of agent they face, thus enabling indirect reciprocity.

Another avenue that has been explored draws inspiration from cultural evolution models. In [156], Hales and Arteconi proposes an evolutionary scheme where cooperation can emerge if both strategies and interaction links can be reliably observed and copied between peers. This behaviour can sustain co-

operation without requiring shared histories, and thus is proposed as an alternative to indirect reciprocity.

In [178], Jian and MacKie-Mason analyse sharing in peer-to-peer networks using two distinct models: direct private incentives for the private provision of public goods, and generalised reciprocity as a strategy in a repeated game. They model generalised reciprocity instances as contribution cycles on a directed graph (in a manner identical to [145]), and formulate a stage game where strategies correspond to sharing vectors. They propose some initial conditions that predict the existence of Nash equilibria with sharing and free riding, even if no direct private incentives are present. These owe their existence to generalised reciprocity and the application of a local *Grim Trigger* strategy. When the model in [145] has been used in experimental studies of indirect reciprocity in humans, it has been found to yield cooperation and increased social benefits in small networks or networks with a high degree of trust between their members. Another similar system is [32], for which Anagnostakis and Greenwald propose an indirect reciprocity scheme based on n-way cyclical exchanges. They propose an algorithm in which a request tree is formed, and when requests that connect disjoint branches of the tree, a cycle is identified through the common ancestor of both branches. Anagnostakis and Greenwald show that the scheme provides superior performance to direct reciprocity.

2.8 Peer-to-Peer Streaming Incentives

The incentive mechanisms that have been discussed in the previous sections of this chapter were not explicitly designed for real time streaming systems. When considering incentive mechanisms for streaming, the real-time nature of the interaction must be necessarily considered, as the incentive mechanism is usually heavily connected to the resource allocation techniques that the system uses - which are very different in streaming overlays from those used in other kinds of overlays.

As a first example, in [296] Tan and Jarvis consider a system in which each stream is divided into substreams, which are then distributed using application-level multicast trees. These trees are built by using an iterated, sealed-bid, first-price auction procedure, where peers participating in substream distribution pick the highest bids and notify the losing bidders of the identities of the winners, so that they can contact them and bid for the substreams again. Tan and Jarvis propose a *Balanced Tree-Shortest Path* strategy that attempts to balance peer resource contribution and overlay network delay minimisation. Further, Tan and Jarvis analyse the differing incentives of stream consuming peers (delay and loss minimisation) and non-consuming, stream-participating peers (currency accumulation maximisation). In this case, the QoS provided by peers becomes a commodity, priced through the auction process. The system does allow deferred consumption of contributions, but is vulnerable to both sybil and whitewashing attacks.

In [155], Habib and Chuang propose the use of rank-order tournaments[26] as a basis for an incentive mechanism scheme to provide service quality differentiation through peer selection in peer-to-peer streaming. The relative contribution of a peer is used as a signal that affects the strategic QoS decisions of the others. Habib and Chuang base their study on PROMISE [163], and use PlanetLab [16] to conduct experiments over the wide-area Internet. They confirm experimentally that peers have a strong disincentive to share their upload capacity while they are downloading (Habib and Chuang do not explain this through a theoretical model, but some possible explanations can be found in [115]). Additionally, they confirm that random peer selection is tantamount to provisioning random QoS, and thus makes it impossible to give any kind of playback guarantees to the destination peer's codec. Habib and Chuang propose a score-based reputation management mechanism to foster cooperation through indirect reciprocity by image scoring [243], propose utility functions for the peers, and then formulate the best response dy-

[26] There is a established research current in economics of using rank-order tournaments as incentive devices to elicit agent effort in principal-agent settings [80] (see Chapter 6 for a description of the principal-agent model).

namics of the system in such a way that they correspond to the best response dynamics of rank-order tournaments. The rationale of the incentive mechanism is that a given peer will refuse to give service to any peer that has a smaller cooperation score than itself - peers only cooperate with those peers that have cooperated at least as much as themselves. Thus, predictably good peers (those that have largest scores) have much greater flexibility in choosing their peers, and thus are capable of obtaining better QoS. Finally, Habib and Chuang compare the effectiveness of their proposed mechanism to the effectiveness of FEC (Forward Error Correction) for codec-level packet loss, and find that up to 35% of FEC overhead is required to have the same loss performance as the incentive mechanism. The system allows deferred benefit from contributions and is resistant to whitewashing attacks, but is vulnerable to sybil attacks.

Some incentive mechanisms for streaming involve the driving of a preference queueing system using incentives information. One instance of this is [248], where Pai and Mohr present an incentive mechanism for peer-to-peer streaming based on the iterated *Prisoner's Dilemma* that gives better QoS to those peers who contribute more to the system. The system only requires direct reputation and reciprocity. Pai and Mohr propose a "Token Stealing Algorithm" in which each peer n_i has a system of token buckets linked to trade accounts (this is similar to GNUNet [149, 47]), and the uploads that other peers have made to n_i are used to drive their own private token buckets, essentially converting upload bandwidth into reserved capacity. This amounts to the implementation of a currency-based market using pairwise currency accounts, and linking past contributions with expedited queueing. From another point of view, this system is essentially a way to implement a currency-based market system using pairwise currency accounts, and linking fragment value to their treatment in a QoS-aware queue. In the words of the authors:

> *The Token Stealing algorithm is a simple extension of the token bucket algorithm. In this algorithm, every peer maintains a standard token bucket that we refer to as the* ***shared bucket*** *into which tokens are added periodically. In addition, the peer maintains a separate bucket for each of its neighbours. We refer to these as* ***private buckets****. Whenever a peer receives a packet from one of its neighbours, it removes tokens from the shared bucket and transfers them to that neighbour's private bucket. This has the effect of* ***reserving*** *a portion of the peer's upload bandwidth to repay the neighbour for the packets it has uploaded. To prevent neighbours from reserving large amounts of bandwidth that they never utilise (for example, because they are connected to other peers with large upload capacities), there is a limit on the size of the private buckets. Tokens that overflow the private buckets are returned to the shared bucket.*
>
> (quote from [248]; emphasis added)

Thus, in a resource-rich system, all peers receive comparable quality of service, whereas in a resource-starved system peers that have contributed (uploaded) to a given peer receive preferential treatment. The system, as it stands, does not accommodate indirect reputation and reciprocity. Furthermore, it only supports direct reciprocity, allowing time deferred consumption only in a very limited fashion.

One of the main problems associated with current peer-to-peer streaming systems is that peer uptime can be short [316, 256, 33]. Thus, making the distribution overlay resilient to peer transience is a key challenge. In [247], Padmanabhan et al. propose CoopNet, a peer-to-peer video streaming system where stream reliability is increased by introducing spatial network diffusion redundancy, as well as MDC[27].

[27] Multiple Description Coding[142] is a coding technique where a single media stream is separated into $n > 1$ independent substreams (*descriptions*). Each description is then divided into packets and routed over uncorrelated, mutually disjoint end-to-end paths. Although any combination of descriptions can be received and decoded, the quality of the reconstructed signal is proportional to the number of recovered descriptions: the more descriptions recovered, the lower the distortion of the original signal. Thus, network QoS problems such as congestion or loss will not interrupt the signal playout, as they would have to affect

CoopNet supports self-interested peers, as their participation in the distribution of media streams is only required when they are actually consuming the stream, and even then, only uploading as much as they download. However, Padmanabhan et al. do not address the issue of freeloading, as they do not propose any kind of enforcement or retaliatory behaviour for the peers. CoopNet is not based on swarming, but on Application Level Multicast (ALM). However, it tries to overcome the main resiliency problem of ALM (the fragility of the distribution tree to failures, peer disconnections or QoS problems at peers close to the root) by using, instead of a single multicast tree, a set of n uncorrelated multicast trees, each carrying an independent substream (MDC description).

The proposed algorithm proceeds as follows. First, the original video stream is encoded as $n > 1$ separate substreams (*descriptions*), after first dividing the stream into *frame groups*. Sections of these frame groups are further classified according to their importance for stream reconstruction, and given commensurate priority and FEC protection. This is done using network loss feedback, in the form of a probability distribution $p(m)$ of successfully receiving m out of the original n substreams. After the n substreams have been generated, they are distributed over the network by using a set of n independent, application layer multicast trees. The creation and maintenance of these multicast trees is one on the main results of [247]. In particular, the following design criteria were used:

- **Short (Bushy) Trees**: The number of application layer hops between the source and the leaves should be minimal. In addition to minimising stream delay, this would minimise the probability of experiencing network problems along the application layer path between the source and the leaves.
- **Tree Diversity**: Each peer should have a different set of ancestors on each of the distribution trees for each description. Thus, any single node failure or disconnection will only impair the transmission of a single description.
- **Tree Efficiency**: All the trees, while maintaining diversity, should be as close to the actual network topology as possible, to minimise propagation delay and unnecessary exposure to network layer congestion and loss.
- **Fast Bootstrapping**: The processing of new peers joining the overlay should be quick, so that the peer starts receiving streaming content as soon as possible.
- **Short Peer Lifetimes**: The reconfiguration of application level multicast trees should be as brief as possible after a given peer has left the overlay or become unresponsive.
- **Scalability**: The tree management algorithm should be able to cope with very large and dynamic trees, in the region of 1000 node arrivals and/or departures per second. Padmanabhan et al. did not elaborate on the memory requirements of their technique.
- **Balance**: The tree should be as balanced as possible.

To manage tree changes as quickly as possible, Padmanabhan et al. considered a centralised tree management algorithm in which all decisions regarding tree constructions were made by the stream source. Furthermore, Padmanabhan et al. concede that this constitutes a performance and reliability single point of failure, but argue that if the source peer fails, a failure on the tree management functions is a moot point: the stream cannot continue anyway.

Thus, when a new peer wishes to join the overlay, it contacts the stream source, which directs it to a designated parent node on each subtree (Padmanabhan et al. do not approach the problem of peer

all descriptions simultaneously. Instead, the reconstructed signal will exhibit transitory episodes of reduced quality. Roughly, the quality of the reconstructed signal will be proportional to the sustained data rate recovered at the receiver.

disobedience or strategic lies). These peers are elected in a deterministic fashion, following the idea (originally proposed for SplitStream [65]) that the outgoing bandwidth constraints for every peer can be honoured by making each peer an interior node in *only one* of the MDC trees, and a leaf on the rest. This also contributes to increase tree diversity (hence reliability) and is the basis for an algorithm proposed by Padmanabhan et al. to build bushy trees.

To make the trees efficient, Padmanabhan et al. propose that, when a number of peers are equally suitable parents for a new one, it should be assigned to the one that is close in terms of QoS or network distance.

In [76], Chu et al. propose a scheme in which resource rich peers contribute excess resources to the system, and subsidise resource-starved peers in order to enable improved social welfare. The protocol is enforced by the media publisher. The media stream is divided into sub-streams, each one with its own multicast tree. Since peers join as many trees as their contributed upload entitles them to, they can strategically decide the best way to use their upload capacity. The system does not support deferred consumption, nor resilience against whitewashing or sybil attacks.

2.8.1 Streaming Resource Depletion and Direct Reciprocity

Streaming systems that use direct reciprocity as an incentive mechanism can suffer from counterproductive chunk diffusion dynamics, because they present skewed fragment availability. We explore this problem below, following the exposition of [248].

We imagine two peers n_i and n_j that have roughly the same delay from the source n_s and are trading fragments with one another. The precise amount each of them downloads from the other will depend on many factors, including their past history of interaction, the current network conditions, and the amount of overlap between the fragments that each one wants and the fragments that the other one has. Now, say that for some reason n_j has been downloading comparatively more fragments from n_i than from other equivalent peers, because n_i has a greater number of fragments that n_j needs.

If n_j were to apply the *Tit-for-Tat* policy to n_i, it would try to preferentially upload to it. However, this would mean uploading to n_i fragments that it does not yet have (n_i is not interested in downloading fragments that it already has), and doing so with small enough delay. Thus, n_j will help n_i to increase its fragment diversity further, and thus the probability of some other peer n_k of downloading preferentially from it will increase. As this continues, more and more peers will aggressively compete for n_i's resources, decreasing its average QoS and concentrating more and more fragments in n_i rather than distributing them around the network. This self-reinforcing process will tend to concentrate a great number of fragments in a single, very contested peer (n_i). This positive feedback loop will tend to amplify small differences between peers, so that peers with small delay or topology advantages will become increasingly important, and other peers will invest large amounts of resources in trying to upload to them while concentrating in them most of their downloads. Thus, general fragment availability will suffer, while the distribution topology becomes more and more dependent on these hubs for connectivity.

In [248], Pai and Mohr tried the opposite solution: reducing the upload rate to those peers that are preferentially downloading from a given peer. This worked in simulations, but had a huge negative impact on system performance in practice. This happens because it propagates the bandwidth limitations of "leaf" peers upstream, so that slow peers that are down the distribution tree tend to slow down peers that are situated further upstream. The net result is that "it was not possible to accommodate the slowest peers without dragging down the performance of the entire system" [248].

Thus, in order for direct reciprocity to be useful for near real time media streaming, contributions must be made in the processes of finding and using chunks. A possibly fruitful direction of research in this direction is the use of random network coding [123] to eliminate the need for fragment search.

2.9 Incentive Mechanisms and Mechanism Design

The reader is referred to Section 3.3 for the definition of some of the technical terms following. Although many of the works detailed before use either game theory or mechanism design as a tool for the formal analysis of the presented incentive mechanisms, there are some works in the literature that have remained closer in spirit to pure applications of game theory or mechanism design. We now present a sample of these works.

2.9.1 Game Theory and Resource Allocation

As a first example, in [31], Altman et al. survey different models and solution concepts for network resource allocation problems in the context of game theory. Equilibrium concepts are analysed, including both Nash-Pareto equilibria [237, 236] and Wardrop Equilibria [314]. Other applications of game theory in communications networks involve the analysis of Braess' Paradox [58], equilibrium control through pricing or mechanism design, equilibrium convergence and Stackelberg frameworks for customer-operator interactions. Practical applications of these models include the analysis of network service provisioning, routing and static/dynamic flow control.

In [140], Gollapudi et al. explore the routing and bandwidth allocation problem by considering each end-to-end flow as a game player attempting to maximise its utility on a routing game. Thus, by considering selfish, rational flows, Nash equilibria are found that simultaneously optimise throughput and fairness criteria. By maximising their own benefits, network flows achieve highly efficient global outcomes. The key technique to accomplish this, rewarding flows in proportion to the benefit they bring to the global outcome, is almost identical to the incentive payment scheme proposed in VCG mechanisms [306, 81, 150]. The issue of finding Nash equilibria is computationally very hard, with no known polynomial-time algorithms [249, 242]. Thus, Gollapudi et al. propose a local search algorithm based on agent best response dynamics, that is simple to implement but still converges to a good approximate equilibrium in a short amount of time.

2.9.2 Mechanism Design and Computer Networks

There is growing interest in the use of mechanism design for the solution of networking problems. In [241], Nisan and Ronen propose the use of mechanism design for the solution of optimisation problems in computing and networking, and gives this technique the name of *Algorithmic Mechanism Design* (AMD). By applying the VCG mechanism (see Chapter 3), Nisan and Ronen show that the shortest path routing problem can be solved even when links are under the control of strategic entities, and then proceeds to formulate and solve a task scheduling problem where k tasks are assigned to n agents in such a way that the total process time is minimised.

The central objection that has been placed on mechanism design for the solution of network problems is the presence of the "centre" that takes agent messages and defines the mechanism outcome and the payments that will be imposed on players. To address this, in [111], Feigenbaum and Shenker argue that a new approach to mechanism design is needed that takes into account not only its algorithmic and complexity implications (like [241]), but the possibility of implementing its proposed solutions in a distributed, scalable fashion. Thus, they posit that extending the notion of AMD to include elements of distributed systems (thus proposing DAMD, *Distributed Algorithmic Mechanism Design*) can not only be the starting point to formulate and solve fundamental open questions such as the notion of *network complexity*, but also to set the foundations of distributed computing with strategic agents. Some applied theoretical issues that Feigenbaum and Shenker propose include using approximation to produce socially near-optimal, strategyproof, near budget-balanced equilibria or the use of cryptography to enable peer accountability. Finally, Feigenbaum and Shenker propose a set of possible application problems, that include web caching, peer-to-peer file sharing, application-layer overlay networks, and distributed task

allocation. A similar, earlier paper is [240], where Nisan considers DAMD and its relationship with conventional mechanism design, and several example problems in resource allocation, optimisation and task scheduling are presented.

In [276], Sanghavi and Hajek study the production of a public resource, where economic actors have an incentive to misrepresent their utilities from the public good in order to minimise their individual contributions. Sanghavi and Hajek assume the existence of a central authority that controls the production and allocation of the the shared resource, as well as the contributions of all peers. However, they propose that the peers collapse their utility functions as a single scalar, as opposed to a whole real valued function as required in the VCG mechanism. This alleviates the communication overheads of the VCG implementation, while retaining good efficiency at its static Nash equilibrium. The system is guaranteed to converge to the Nash equilibrium if the peers follow their best response dynamics (the authors do not study arbitrary non-myopic strategies).

More recent works, such as [283], reinforce the general position of the research community with respect to the usefulness of game-theoretical approaches, and propose a set of open research questions that need to be addressed by AMD/DAMD. Since only strategic (rational or irrational) peers are susceptible to incentives, the goal of DAMD is to set up an incentive scheme that allows irrational peers to become rational, and rational peers to behave in a way that attains a good enough social welfare level.

2.10 Overlay-ISP Interaction for Managed Overlays

If one considers the edge-to-edge traffic pattern most useful to a given peer-to-peer system or managed overlay, it is clear that it will be a function of the preferences of the overlay peers regarding QoS, resource availability and data caching and replication. On the other hand, if one considers the edge-to-edge traffic pattern most desirable to the ISP underlying it, it will be a function of its link and node costs, the background traffic that it carries and its traffic engineering policies. Thus, a tension between the preferences of the overlay and the ISP arises and can be analysed from a game-theoretical perspective [31].

This tension has been very visible in the context of net neutrality [93], and it has been accompanied by tools such as Switzerland [25], Glasnost [23] or *bttest* [104] that can be used to detect the *bandwidth throttling* phenomenon that emerges as a rational strategy for ISPs [312]. Proposals to shift this equilibrium away from throttling and into more overlay-friendly states include bandwidth auctions [322] and infrastructure caching, if the ISP can attract a sufficiently large number of subscribers [133].

Regarding the analysis of cooperative equilibrium outcomes between ISPs and managed overlays, work has been done through the use of the Shapley value of coalitional games [211, 212], and there is a long tradition of work regarding incentives issues in routing - see, for instance [139, 202, 43, 259, 113, 128, 63, 193]. Finally, it is important to note that there is currently a large interest in the standardisation of interface to guide the equilibrium outcomes of these overlay-ISP interactions [252], and an increased interest in the research community in exploring the issues behind the interaction of routing and content distribution [103, 180, 184, 211, 28].

In Chapter 7, we present a model for the utility-maximising behaviour of managed overlays and the profit-maximising behaviour of ISPs.

Part II

Theory and Contributions

3
Underlying Theoretical Models

3.1 Graph Theory

Almost all contributions in this work refer to networks, and thus a basic set of graph-theoretical definitions is fundamental. We follow nomenclature similar to that of [148].

3.1.1 Nodes and Links

Definition 3.1. A *directed graph* $\mathcal{G}$ as a finite set of vertices, or *nodes* ($\mathsf{N} = \{n_i\}$) and a set of edges, or *links* ($\mathsf{L} = \{l_i\}$) defined over the cartesian product $\mathsf{N} \times \mathsf{N}$: each link l_i can thus be seen as a relationship between two nodes. As the presence or absence of a link between two given nodes limits the ways in which the network is traversable, it defines its *topological structure*.

The number of nodes in the network is $|\mathsf{N}|$, the cardinality of N. The number of links in the network is, equivalently, $|\mathsf{L}|$.

Links correspond to ordered node pairs and have an intrinsic directionality: one of endpoints of the link is designated as the *start*, and the other as the *end*. The link is thus directed from its start to its end. We will use the notation $start(l_i)$ and $end(l_i)$ to denote, respectively, the start and end of link l_i.

Definition 3.2. A link l_k is called *outgoing* from node n_i, expressed as $n_i \rightarrow l_k$, if it originates on n_i ($start(l_k) = n_i$). Equivalently, a link l_k is called *incoming* to node n_j, expressed as $l_k \rightarrow n_j$, if it terminates on n_j ($end(l_k) = n_j$).

Definition 3.3. A link is *incident* on a node if it is either outgoing from or incoming to it.

Definition 3.4. Two nodes are *adjacent* if there is a link between them.

Definition 3.5. Two links are *adjacent* if they have an endpoint in common.

Definition 3.6. The *incoming link neighbourhood* $\Lambda^+(n_i)$ of a node n_i is a set consisting of all incoming links terminating on the node. Thus, the *indegree* of n_i, $|\Lambda^+(n_i)| = \deg^+(n_i)$, is the number of incoming links incident on it.

The incoming link neighbourhood $\Lambda^+(\mathsf{S})$ of a connected subgraph S in G is the set consisting of all incoming links terminating on any node of S and originating on any node outside S.

Definition 3.7. The *outgoing link neighbourhood* $\Lambda^-(n_i)$ of a node n_i is a set consisting of all the outgoing links originating on the node. Thus, the *outdegree* of n_i, $|\Lambda^-(n_i)| = \deg^-(n_i)$, is the number of outgoing links originating on it.

The outgoing link neighbourhood $\Lambda^-(\mathsf{S})$ of a connected subgraph S is the set consisting of all outgoing links originating on any node of S and terminating on any node outside S.

Definition 3.8. The *link neighbourhood* $\Lambda(n_i) = \Lambda^-(n_i) \bigcup \Lambda^+(n_i)$ of a node n_i consists of all links incident on the node. Thus, the *degree* of n_i, $|\Lambda(n_i)| = |\Lambda^+(n_i)| + |\Lambda^-(n_i)| = \deg(n_i) = \deg^+(n_i) + \deg^-(n_i)$, is the number of both incoming and outgoing links incident on it.

The link neighbourhood $\Lambda(\mathsf{S})$ of a connected subgraph S is the set consisting of all links with exactly one endpoint in S.

Definition 3.9. The *incoming node neighbourhood* $\Pi^+(n_i)$ of a node n_i is the set of nodes adjacent to n_i through the links in $\Lambda^+(n_i)$.

The incoming node neighbourhood $\Pi^+(\mathsf{S})$ of a connected subgraph S is the set consisting of all nodes adjacent to any node in S through a link in $\Lambda^+(\mathsf{S})$.

Definition 3.10. The *outgoing node neighbourhood* $\Pi^-(n_i)$ of a node n_i is the set of nodes adjacent to n_i through the links in $\Lambda^-(n_i)$.

The outgoing node neighbourhood $\Pi^-(\mathsf{S})$ of a connected subgraph S is the set consisting of all nodes adjacent to any node in S through a link in $\Lambda^-(\mathsf{S})$.

Definition 3.11. The *node neighbourhood* $\Pi(n_i)$ of a node n_i is the set of nodes adjacent to n_i through the links in $\Lambda(n_i)$.

The node neighbourhood $\Pi(\mathsf{S})$ of a connected subgraph S is the set consisting of all nodes adjacent to any node in S through a link in $\Lambda(\mathsf{S})$.

3.1.2 Walks and Paths

Definition 3.12. A *walk* W_{nm} between two nodes n_n and n_m is an alternating sequence $W = n(0), l(1), n(1), l(2), \ldots, n(k-1), l(k), n(k)$ of nodes and links, such that $l(i+1)$ is incident on $n(i)$ and $n(i+1)$. Furthermore, $n(0) = n_n$ and $n(k) = n_m$.

Definition 3.13. A *trail* B_{nm} is a walk W_{nm} with no repeated links, that is, where $l(i) = l(j)$ if and only if $i = j$. A *directed trail* is a trail where, for each link $l(i)$, a *direction* is defined such that if $n(i-1) \rightarrow l(i) \rightarrow n(i)$ the direction is **positive**, and if $n(i-1) \leftarrow l(i) \leftarrow n(i)$, the direction is **negative**. Intuitively, this means that if the direction of a link on a trail matches the direction implied by its node sequence, the edge is *positively directed*, whereas if its direction is opposite to the direction implied by its node sequence, the edge is *negatively directed*.

Definition 3.14. A *path* P_{nm} is a trail with no repeated nodes unless possibly the initial node n_n and the final final node n_m. Thus, $n(i) = n(j)$ if and only if $i = j$, except for the first and last nodes, where $n(0) = n(k)$ is allowed (in this case, the path is a *cycle* and $n = m$). Again, a *directed path* is a cycle where direction is defined in the same way as in directed trails.

In Chapter 4 we will be interested in the analysis of all the possible paths between two nodes in a directed graph. The following definitions will simplify the exposition.

Definition 3.15. The *graph power* $\mathcal{G}^n$ of degree $n \geq 2$ of a graph $\mathcal{G}^1 = (\mathsf{N}^1, \mathsf{L}^1)$ is a graph with the same N as $\mathcal{G}^1$, but a different link set L^n, so that there is a link between n_i and n_j in L^n only if there is at least one path P_{ij} in $\mathcal{G}^1$ of n hops or fewer between n_i and n_j.

Definition 3.16. The *node diameter* $d(\mathcal{G}, n_i)$ of a graph $\mathcal{G}$ centred on a node n_i is the minimum number of hops h so that all nodes in $\mathcal{G}$ can be reached from n_i in at most h hops.

Definition 3.17. The *graph diameter* $d(\mathcal{G})$ of a graph $\mathcal{G}$ is the maximum node diameter of its constituent nodes. Formally, $d(\mathcal{G}) = \max_{n_i \in \mathsf{N}} d(\mathcal{G}, n_i)$.

Definition 3.18. The *maximum graph power* $\mathcal{G}^{d(\mathcal{G})}$ of a graph $\mathcal{G}$ is the n-th power of the graph for $n = d(\mathcal{G})$. Thus, there is a link between n_i and n_j in $\mathcal{G}^{d(\mathcal{G})}$ if there is at least one path in P_{ij} in $\mathcal{G}$ between n_i and n_j.

3.1.3 Cycles

Definition 3.19. An *oriented cycle* is a cycle that is a directed path as well. Thus, orientation on a cycle defines a "sense of rotation" along the cycle, and each of the links in the cycle can be *positively* or *negatively* directed.

Definition 3.20. A link l_i within a cycle c_k is *positively directed*, denoted as $l_i \circlearrowleft c_k$, if the direction of l_i matches the "sense of rotation" along c_k.

Definition 3.21. A link l_i within a cycle c_k is *negatively directed*, denoted as $l_i \circlearrowright c_k$, if the direction of l_i opposes the "sense of rotation" along c_k.

Definition 3.22. Two cycles are *adjacent* if they share at least one link.

3.2 The Mathematics of Sybil Attacks

In the following we provide a theoretical basis for the analysis of the sybil attack.

3.2.1 Whitewashing and Sybil Attacks

An underlying assumption of any reputation system is that a peer can be reliably identified over time. Thus, reputation systems are usually susceptible to identity-based attacks. The main three attacks that have been addressed in the literature are *slander*, *whitewashing* [117] and *sybil attacks* [106].

In most reputation systems, peers receive reports from other peers regarding the behaviour and contributions of other peers in the trust network. Thus, peers can create reports with false information regarding other peers, send reports pretending to be other peers, or arbitrarily modify reputation system messages as they traverse the network - and this might be difficult to detect. One particular way in which peers can lie or modify passing messages is to artificially diminish the contributions or reputation of a victim. This attack, known as *slander*, depends on the ability of peers to distinguish true information with respect to a given peer from lies planted by other peers. This problem has been approached from two main angles: using anomaly detection [183] and relying on statistical averaging to dilute the effect of attackers [147].

Another related attack is *whitewashing*, that occurs when a peer can dump an identity that has been targeted as untrustworthy by the reputation system and just use a new one. The whitewashing nodes are then indistinguishable from legitimate newcomers, and can exploit any kind of allowances given to new nodes to help them bootstrap into the system. Thus, whitewashing leverages the extremely low marginal cost that creating new identities has in current peer-to-peer systems to destroy the sanctioning capabilities of reputation systems, and complicates the insertion of new peers into the peer-to-peer substrate in the process.

A third attack is the *sybil attack*. A Sybil attack occurs when a user creates a large (potentially unbounded) number of identities that then collude towards the user's aims. Not only can these identities be created at almost no cost, any sort of past contribution history can be simulated as well.

Conventionally, reputation systems operate by assuming a correspondence between peer names and their identities, and thus they are usually susceptible to sybil attacks [105]. Although these problems are still open, numerous countermeasures have been proposed (see [203] for a general survey, and [323] for recent work in leveraging social network structure for the prevention of sybil attacks). Even though eliminating the scope for sybil attacks may be very difficult within the current paradigms of reputation systems [106], even imperfect identity management can decrease freeloading to a manageable level. For instance, Marti and Garcia-Molina show that having a basic reputation system based on self-signed certificates can provide large performance improvements (4 to 20-fold, as reported in [219]) on a file-sharing peer-to-peer network against node selfish behaviour, when compared with no reputation system at all.

3.2.2 Definition of Sybilproofness

In this Section we will be concerned with the *trust network* of a peer-to-peer overlay: a weighted, directed graph $\mathcal{G}$ (for nomenclature see Section 3.1) where the link weights $w_{ij} \geq 0$ associated with the links l_{ij} correspond in some way with the trust that peer n_i has in peer n_j. The manner in which these link trust values are obtained is not of concern for the material presented. In Chapter 4, which is the main driver for this Section, $\mathcal{G}$ is identified with past contributions between peers, and w_{ij} is just a record of unreciprocated contributions backed by reciprocative policies.

Sybilproofness was presented in [71] (which then became Chapter 27 of [242]) in the context of reputation systems. We respect the original nomenclature by Cheng and Friedman in [71].

Definition 3.23. A *reputation function* $f(\mathcal{G})$ is a function $f : \mathsf{N} \mapsto \mathbb{R}$, where N is the node set of a graph $\mathcal{G}$. The value that f assigns to node n_i will be denoted as $f(\mathcal{G}, n_i)$.

A *sybil strategy* is the analytical representation of a sybil attack. Since a peer can create not only arbitrary identities, but also any relationships between them, a sybil strategy is comprised of an arbitrary number of nodes and trust links, forming an arbitrary topology. Formally, we have that:

Definition 3.24. A *sybil strategy* for a malicious peer n_m is a graph $\mathsf{S}_m = (\mathsf{N}_m, \mathsf{L}_m)$ along with a set $\Lambda(\mathsf{S}_m)$ of *attack links*. Of course, N_m and L_m can be arbitrarily large, and they can form an arbitrary topology. This $(\mathsf{N}_m, \mathsf{L}_m)$ topology connects to $\mathcal{G}_{-m}$, the rest of the trust network, through the set of *attack links* $\Lambda(\mathsf{S}_m)$ which could be potentially very large. Each link in L_m is annotated in an arbitrary way, describing any claimed past history or trust between the sybil nodes. Links in $\Lambda(\mathsf{S}_m)$ are annotated using normal trust building procedures with their correspondent peers.

We explore the concept of sybil strategy further. Let $\mathcal{G} = (\mathsf{N}, \mathsf{L})$ represent the true state of a trust network, and $n_m \in \mathsf{N}$ a malicious node mounting a sybil attack from within $\mathcal{G}$. Let $\mathcal{G}' = (\mathsf{N}', \mathsf{L}')$ be the trust network that results if we include the sybil strategy S_m of n_m in $\mathcal{G}$, so that S_m is the connected subgraph of $\mathcal{G}'$ that constitutes the sybil attack by n_m. Since the links between sybils and real nodes must be created following true trust building processes, we have that $\Pi(\mathsf{N}_m) \in \mathsf{N}'$ is identical to $\Pi(n_m) \in \mathsf{N}$ and, if we collapse $\mathcal{G}_m$ to its only real node n_m, $\mathcal{G}'$ is transformed into $\mathcal{G}$. Of course, we have that $\mathsf{N}' = \mathsf{N} \cup \mathsf{N}_m$ and $\mathsf{L}' = (\mathsf{L} \setminus \Lambda(n_m)) \cup \mathsf{L}_m \cup \Lambda(\mathsf{S}_m)$, since the attack substitutes the links between n_m and its adjacent nodes with the attack links.

Definition 3.25. As before, let $\mathcal{G} = (\mathsf{N}, \mathsf{L})$ represent an arbitrary trust network, and $\mathcal{G}' = (\mathsf{N}', \mathsf{L}')$ be the trust network including the *sybil strategy* $\mathcal{G}_m = (\mathsf{N}_m, \mathsf{L}_m)$. A reputation function f is *value sybilproof* if,

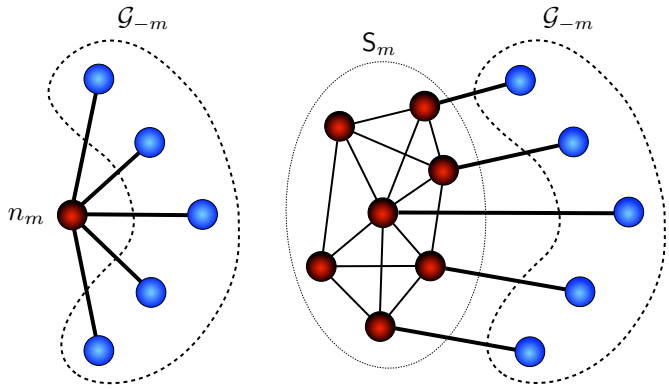


Figure 3.1: A malicious node n_m and its sybil strategy S_m. The links between n_m (or S_m) and the rest of the network, $\mathcal{G}_{-m}$, are the *attack links*.

for every $\mathcal{G}$ and peer $n_m \in \mathsf{N}$, there is no sybil strategy $\mathcal{G}_m = (\mathsf{N}_m, \mathsf{L}_m)$ such that $f(\mathcal{G}', n_s) > f(\mathcal{G}, n_m)$ for any sybil node $n_s \in \mathsf{N}_m$.

Early reputation systems like *EigenTrust* [185] relied on symmetric reputation functions, which depend only in the topology of the trust network and are therefore invariant when *symmetry transformations* are applied to the trust network. It is easy to prove that these systems can not be sybilproof, but we must define some concepts first that will allow us to discuss network symmetries more easily.

Definition 3.26. A *graph isomorphism* from a graph $\mathcal{G} = (\mathsf{N}^G, \mathsf{L}^G)$ to a graph $\mathcal{H} = (\mathsf{N}^H, \mathsf{L}^H)$ is a function $\chi : \mathsf{N}^G \longrightarrow \mathsf{N}^H$ that assigns a node in $\mathcal{H}$ to each one of the nodes of $\mathcal{G}$ while maintaining adjacency. Thus, if n_i and n_j are adjacent in $\mathcal{G}$, $\chi(n_i)$ and $\chi(n_j)$ must be adjacent in $\mathcal{H}$.

In practice, we say that two graphs $\mathcal{G}$ and $\mathcal{H}$ are *isomorphic* if it is possible to re-label the nodes of one to yield the other. A symmetric reputation function can now be defined:

Definition 3.27. Let $\mathcal{G} = (\mathsf{N}^G, \mathsf{L}^G)$ represent a trust network, and χ an isomorphism of $\mathcal{G}$ such that $\mathsf{N}^H = \chi(\mathsf{N}^G)$, and $\mathcal{H} = (\mathsf{N}^H, \mathsf{L}^H)$ is any graph isomorphic to $\mathcal{G}$. A reputation function f is *symmetric* if we have that $f(\mathcal{G}, n_m) = f(\mathcal{H}, \chi(n_m))$.

Thus, a symmetric reputation function assigns values on the nodes only taking into account the trust topology: their position in the trust network $\mathcal{G}$ and the weights of its links. Although attractive mathematically, this property is of very limited value in a trust network in which sybils can be introduced: all peers can use sybils to implement any topology with any link weights.

A basic conclusion of [71] is that symmetric reputation functions are never sybilproof. The proof of this result is simple, and relies on the fact that an automorphism is an isomorphism.

Definition 3.28. A *graph automorphism* of a graph $\mathcal{G} = (\mathsf{N}, \mathsf{L})$ is a graph isomorphism to itself. Thus, if $\hat{\chi}$ is an automorphism of $\mathcal{G}$, it re-labels nodes in N in such a way that the adjacencies of $\mathcal{G}$ are preserved. Thus, any automorphism $\hat{\chi}$ of $\mathcal{G}$ is a permutation on N that maintains the adjacency structure of $\mathcal{G}$ - if two nodes are joined by an edge in N, so are their images under the permutation $\hat{\chi}(\mathsf{N})$.

Of course, every network has at least one automorphism: the identity automorphism χ_I, that maps every node to itself. We call this a *trivial* automorphism, and explicitly remove it from our analysis. We shall restrict our attention to nontrivial automorphisms that reflect true network symmetries.

Theorem 3.1. *No symmetric, nontrivial reputation function can be sybilproof.*

Proof. Let $\hat{\mathsf{f}}$ be a reputation function such that not all of N is mapped to the same value ($\hat{\mathsf{f}}$ is *nontrivial*). Consider a sybil strategy S_m that consists of duplicating the entire $\mathcal{G}$ by creating a sybil node for each node in $\mathcal{G}$, also replicating any trust links between them (S_m is isomorphic to $\mathcal{G}$). Then, $\mathcal{G}'$ will consist of two identical copies of $\mathcal{G}$ joined at the attacking node n_m: one of the copies consists of true peers; the other one consists of sybils (the sybil n_m corresponds to itself). Obviously, $\mathcal{G}'$ has an automorphism χ such that every peer is mapped to its corresponding sybil, and by Definition 3.27, n_m will always have at its disposal a sybil that achieves the maximum reputation value in the network (indeed, *any* value in the network). □

3.2.3 Conditions for Sybilproofness

The solution that [71] presents for this problem is *anchoring* reputation values to some peer in the network. After a *trust anchor* n_a has been set, reputation will propagate along outgoing paths from n_a. In the analysis that follows, we present the exposition in [71] and, like [71], we extensively use the notion of a *capacity-edge-disjoint path*. In this formulation, if there is an edge $e = (n_i, n_j)$ with trust value w_{ij}, it is possible to split it into many links $e_1 = (n_i, n_j)$, $e_2 = (n_i, n_j)$, ... with weights w_{ij}^1, w_{ij}^2, ..., as long as the trust value of the link is *split additively*. This would imply that $w_{ij}^1 + w_{ij}^2 + \ldots = w_{ij}$, so that no trust is lost in the process. The basis of this is that a peer mounting a sybil attack will choose to assign its contributions or trust values to any of its sybils in the most advantageous manner, but all of them will ultimately bounded by the total trust between the nodes in the border of the sybil strategy and the rest of the trust network. Figure 3.2 shows an example of this, where a true trust link between peers n_j and n_i (Figure 3.2a) can be expanded into either a set of trust links from n_j to the sybils of n_i (Figure 3.2b), a set of trust links from the sybils of n_j to n_i (Figure 3.2c) or a set of trust links from the sybils of n_j to the sybils of n_i (Figure 3.2d). Although the real trust value w_{ij} could be partitioned in various ways, the total outgoing trust from n_j (and its sybils) to n_i (and its sybils) will be equal to w_{ij}. This is because n_i and n_j are not sybils of a single node, and thus trust between them cannot be arbitrarily created - it must be earned in some way[1].

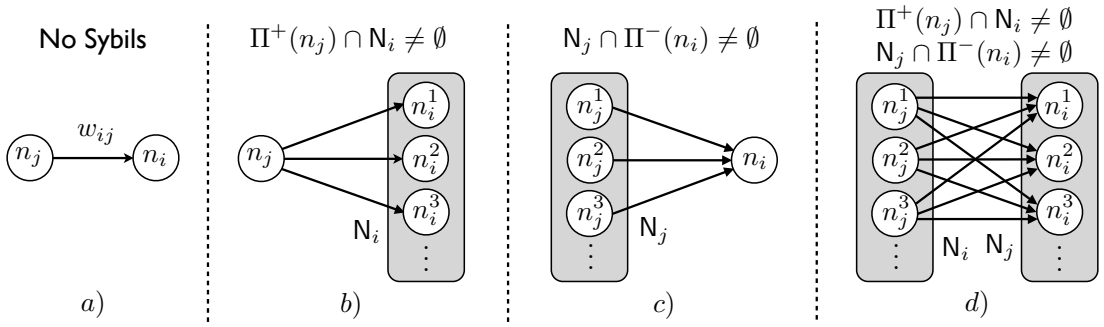


Figure 3.2: Sybil strategies and disjoint paths. N_i is a sybil strategy for n_i and N_j is a sybil strategy for n_j.

Definition 3.29. Let $\mathcal{G} = (\mathsf{N}, \mathsf{L})$ represent a trust network, and n_a the *anchor* node. Further, let P_{ai} be the set of all capacity-edge-disjoint paths between n_a and the node whose reputation is being calculated, n_i. In addition, we define $\mathbb{P}_{ai}$ as $\mathcal{P}(\mathsf{P}_{ai})$ - the *power set* of P_{ai} (the set of all subsets of P_{ai}), so that $\mathbb{P}_{ai}$ is the set of all possible sets of capacity-edge-disjoint paths between the root node n_a and peer n_i. Finally, $g(P_{ai}) : \mathsf{P}_{ai} \mapsto \mathbb{R}$ is a *reputation flow* function that takes the weights of the links in P_{ai} and

[1] We assume that n_j and n_i are not colluding. If they are, they are treated as part of the same sybil strategy and will only be able to interact with the rest of the peers through capacity-edge-disjoint paths.

yields the amount of reputation that can flow from n_a to n_i via P_{ai}. Then, an *anchor-flow* reputation function is defined as

$$\mathsf{f}_a(\mathcal{G}, n_i) = \max_{\mathscr{P}_{ai} \in \mathbb{P}_{ai}} \left(\bigoplus_{P_{ai} \in \mathscr{P}_{ai}} g(P_{ai}) \right). \tag{3.1}$$

It is instructive to decompose the operations that the various components of (3.1) are performing on the fundamental link trust values. $\mathscr{P}_{ai}$ is the set of capacity-edge-disjoint paths that maximises the expression in parenthesis in (3.1), P_{ai} iterates over each one of the paths in $\mathscr{P}_{ai}$, and the function g is applied to obtain the reputation value that n_a provides to n_i through P_{ai}. These values are aggregated over the set $\mathscr{P}_{ai}$ using $\oplus$ to yield the reputation value $\mathsf{f}_a(\mathcal{G}, n_i)$. Thus, g aggregates link weights over paths and $\oplus$ aggregates paths over path sets. The final $\max$ operation aggregates the entire $\mathbb{P}_{ai}$ to one of its elements $\mathscr{P}_{ai}$. Thus, (3.1) is essentially a hierarchical aggregation scheme applying the operators g, $\oplus$ and $\max$.

Having defined $\mathsf{f}_a(\mathcal{G}, n_i)$, it is now possible to state the requirements that an anchor-flow reputation function must have in order to be sybilproof:

1. *Diminishing returns*. For all paths P_{ai}, if path P_{ij} is concatenated to P_{ai} to obtain P_{aj} then $g(P_{ai}) \geq g(P_{aj})$. Thus, if we denote path concatenation with $\cup$, we have that $g(P_{ai}) \geq g(P_{ai} \cup P_{aj})$. As the length of a path for reputation propagation increases, its reputation flow either decreases or remains unchanged. This counters the capacity of creating arbitrarily long chains of sybils with arbitrarily high trust links in them.

2. *Monotonicity*. For all path sets $\mathscr{P}_{ai}$, if another path P_{ai} is added to $\mathscr{P}_{ai}$ to obtain $\mathscr{P}'_{ai} = \mathscr{P}_{ai} \cup P_{ai}$ then $\bigoplus \mathscr{P}_{ai} \leq \bigoplus \mathscr{P}'_{ai}$. This means that $\oplus$ is nondecreasing: as the number of reputation-carrying, capacity-edge-disjoint paths from n_a to n_i increases, the total reputation flow from n_a to n_i either increases or remains unchanged. This property gives $\mathsf{f}_a(\mathcal{G}, n_i)$ its reputation function properties, allowing a higher trust value to be achieved if the reputation flow over $\mathscr{P}_{ai}$ increases. Informally, this can be understood as the trust flow between n_a and n_i increasing with the number of independent transitive trust paths between n_a and n_i (we assume that n_i and n_a are not both sybils of the same peer, as this case is of no interest).

   Furthermore, the *diminished returns* property is extended with the requirement that g is nondecreasing with respect to *edge values*. This means that if the trust value of any link which forms part of path P_{ai} increases, $g(P_{ai})$ must not decrease. Again, this property allows f_a to behave as a reputation function: if the trust value of a link in a path increases, the trust transitivity over that path is in some sense "stronger", and the end-to-end trust should not decrease.

3. *No splitting*. Given a single path P_{ai}, if we split it into two edge-disjoint paths P^1_{ai} and P^2_{ai} by decomposing each of its constituent links into two parallel links , then $P^1_{ai} \otimes P^2_{ai} \leq g(P_{ai})$. This counters the capacity of creating arbitrarily many sybils with arbitrarily many links to peers outside the sybil strategy set.

We use these properties to justify the sybilproofness of a novel contribution/currency distributed accounting mechanism in Chapter 4.

3.3 Game Theory

Game theory [238, 236] is the mathematical modelling of the interactions between strategic agents. Typically, game theory assumes that players in a game are *rational*: not only do they always make the decision that benefits them the most based on the information available (as defined by the game), but also, they are always capable of arriving at that decision regardless of the computational burden which

might be incurred. Branches of game theory include cooperative game theory, in which players coalitions negotiate and enforce bargains, and non-cooperative game theory, in which the only meaningful agreements are those which are "self-enforcing": those for which the game provides the players with an incentive to respect. We now define some of the concepts that we will use in Chapters 5, 6 and 7. For a more detailed introduction to game theory, see [245, 137].

A *game* is a model of the interaction of autonomous agents called *players*. At any given point of the game, each player has a set of *actions* available to it. However, the benefit that a player can eventually obtain from any given action is dependent on the actions of the other players at that point (and all subsequent points) of the game. The game eventually ends with an *outcome* that is a result of the decisions of the players, which can then rank, according to their preferences, all the possible outcomes that can result form all possible combinations combinations of player actions. We now formalise these concepts.

Definition 3.30. We define a *game* G as consisting of:

1. A set $\mathsf{P} = \{\mathsf{p}_j\}$ of *players*. In our case, there will usually be a peer n_i that "hosts" the game, but will not play itself. Thus, we have that $\mathsf{P} = \mathsf{N} \setminus n_i$.

2. A *type* $\zeta_j \in \mathcal{Z}_j$ for each player, where $\mathcal{Z}_j$ the set of all possible types for p_j. The set of all possible types in the game, $\mathcal{Z} = \mathcal{Z}_1 \times \mathcal{Z}_2 \times \ldots \times \mathcal{Z}_{|\mathsf{P}|}$ will be called the *type space* of the game.

   The type of a player parametrises the utility that it obtains for any given game outcome (see *utility* below), and it is usually *private information*: only p_j knows ζ_j - in particular, n_i does not know ζ_j. We call $\zeta = (\zeta_1, \zeta_2, \ldots, \zeta_n) \in \mathcal{Z}$ the *type profile* of the game.

3. A set of possible *actions* $\mathcal{A}_j$ for each player p_j. At each decision point in the game, any player p_j chooses an action $a_j \in \mathcal{A}_j$. The sequence of actions that any given player p_j takes over the duration of a game is called a *strategy* $s_j \in \mathcal{S}_j$. We call $\mathcal{S}_j$ the *strategy space* of player p_j. We define $\mathcal{S} = \mathcal{S}_1 \times \mathcal{S}_2 \times \ldots \times \mathcal{S}_n$ as the strategy space of the game, and each element s of $\mathcal{S}$ as a *strategy profile*.

4. A *game outcome* $o \in \mathcal{O}$, where $\mathcal{O}$ is the *outcome space* of the game. Informally, the outcome is the state of the game when no p_j has any actions remaining and the game ends. We abuse notation by defining a function $\mathsf{o} : \mathcal{S} \mapsto \mathcal{O}$ that maps the chosen strategies of all peers to the game outcome. Thus, the game outcome depends on the combinations of actions that the players choose, and any outcome o_s can be expressed as a function of any given strategy profile: $o_s = \mathsf{o}(s_1, s_2, \ldots, s_n) = \mathsf{o}(s)$ for any strategy profile s.

5. For each player p_j, there is a *preference relation* that models the way that p_j ranks the outcomes in terms of their desirability. This preference relation will be modelled using a *utility function* U_j, so that any given player p_j *prefers* an outcome o_s over another outcome o_t if $U_j(\zeta_j, o_s) > U_j(\zeta_j, o_t)$. Further, we are assume that all $\mathsf{p}_j \in \mathsf{P}$ have *quasilinear preferences*, in the sense that if any agent receives a *payment* of h_j units as part of the game outcome, its utility will be $U_j(\zeta_j, o_s) = f(\zeta_j, o_s) + \mathsf{h}_j$ for some concave f.

The strategic elements of a game outcome are defined by selecting a *solution concept* that identifies the equilibria conditions of the game. The most common solution concept is the Nash equilibrium: a strategy profile where every player selects the strategy that maximises its utility, *taking the strategies of all other players as given*. We define the Nash equilibrium in terms of the *player best response functions*:

Definition 3.31. Let $s_{-j} = (s_1, s_2, \ldots, s_{j-1}, s_{j+1}, \ldots, s_{|\mathsf{N}|}) = s \setminus s_j$ be the strategy profile of all peers with the exception of p_j. A strategy s_j^+ is a *best response* for peer p_j against the strategy profile s_{-j} of all other players if it enforces an outcome o that maximises U_j. Thus, we have that

$$U_j\left(\zeta_j, \mathsf{o}(s_j^+, s_{-j})\right) \geq U_j\left(\zeta_j, \mathsf{o}(s_j, s_{-j})\right) \ \forall s_j \neq s_j^+.$$

A *best response function* for player p_j is a function ϑ_j such that $s_j^+ = \vartheta_j(\zeta_{-j}, s_{-j})$.

In essence, a Nash equilibrium is a strategy profile where no player can increase its utility by unilaterally changing its strategy. Thus, the strategy followed by each of the players is a best response to the strategies being used by all the other players. Formally:

Definition 3.32. A strategy profile $s^\star$ is a *Nash equilibrium* of a game G if the strategy $s_j^\star$ that a given player p_j follows is a best response to the strategy profile $s_{-j}^\star$. Thus, it follows that

$$s_j^\star = \vartheta_j(\zeta_j, s_{-j}^\star) \text{ for every player } \mathsf{p}_j.$$

In a Nash equilibrium, thus, every player chooses its strategy $s_j^\star$ to maximise its utility U_j, given its type ζ_j and the Nash equilibrium strategies $s_{-j}^\star$ of every other player. Although the Nash equilibrium is a natural solution concept for a strategic game, it has the drawback of requiring each player to know all the possible strategies that the other players can use, and the resulting outcome as a function of its own strategies. This makes Nash equilibria difficult to compute (see [242] Chapter 2). We will seek to enforce a stronger class of equilibrium that is easier to compute.

A *dominant strategy equilibrium* is a strategy profile where no player has an incentive to change its strategy, *independently of the strategies being followed by all other agents*. Thus, the strategy that each of the players follows is a best response to all the the strategy profiles that all the other players could play. Formally:

Definition 3.33. A strategy profile s^* is a *dominant strategy equilibrium* of a game G if the strategy that a given player p_j follows is a best response to every possible strategy profile $s_{-j} \in \mathcal{S}_{-j}$, where $\mathcal{S}_{-j} = \mathcal{S}_1 \times \mathcal{S}_2 \times \ldots \times \mathcal{S}_{j-1} \times \mathcal{S}_{j+1} \times \ldots \times \mathcal{S}_{|\mathsf{N}|}$. Thus, *the specific strategy profile that other agents might be using is irrelevant for* p_j *when defining which strategy to use*. Formally, we have that

$$s_j^* = \vartheta_j(\zeta_j) \text{ for every player } \mathsf{p}_j.$$

3.4 Economics of Peer-to-Peer Incentives

3.4.1 Incentives and Unobservability: Hidden Action

The term *hidden action* is used when, in a two party transaction, the actions of one of the parties (usually regarding contract compliance or lack thereof) are hidden from the other. One model for these cases is the *principal-agent* model, where an economic actor (the *principal*) trusts another one (the *agent*) to perform a task. However, the actions that the agent takes (or fails to take) in the context of the task are not completely verifiable by the principal. Thus, the agent has the power to impose, through its actions, an externality on the principal. To transfer some of this risk back to the agent, the principal will usually require a contract by which the value that the the agent obtains from the transaction will depend on the observable consequences of its actions. This contract is designed to transfer at least part of the externality back to the agent. Additionally, this contract creates an "audit point" where the principal is able to assess the actions of the agent, and respond accordingly.

ϕ	Level of effort by the agent
U_s	Utility function for the agent
U_c	Utility function for the principal
q	Project outcome: income for the principal
ψ	Payment given by the principal to the agent
p	Probability of the agent achieving given outcome, as a function of its effort.
U_r	Reservation utility for the agent

Table 3.1: Principal-Agent model notation

3.4.1.1 The Principal-Agent Model

The most natural explanation of the principal-agent model arises in the modelling of project and firm management. Suppose that the owner of a project (the *principal*) hires a manager (the *agent*) to run it. Since the principal will not supply labour to the project, he will only be interested in its pecuniary outcome.

Let ϕ be the level of effort that the agent puts into the project, and ψ the payment that he will receive from the principal after the project is completed. We assume that the agent will have a quasi-linear utility $U_s = f_s(\psi) - \phi$ with f nondecreasing and concave[2]. Since we assume that the agent participates in a labour market, there is a minimum *reservation utility* U_r for the agent: the minimum benefit that he must obtain from the pay-effort combination offered by the project with the principal so that it is rational to participate in the project rather than taking any other employment opportunity. More formally, U_r is the expected utility for the agent in the current labour market. We denote the expected utility that the principal gets from the project as U_c. Again, U_c will be a quasi-linear utility $U_c = f_c(q) - \psi$ where q represents the income that the principal obtains from the project (referred to as the *project outcome*).

The critical feature of the principal-agent model is that *it assumes that the principal is unable to measure the effort level ϕ directly*. Thus, ϕ is always private information to the agent. Moreover, the principal is unable to unambiguously infer ϕ from the project outcome - a number of possible values of ϕ can yield the same project outcome q. The outcome of the project, however, does depend probabilistically upon the level of effort ϕ of the agent. To address the simplest case, let there be two possible project outcomes q_+ and q_- with $q_+ > q_-$ and two possible levels of effort by the agent, ϕ_+ and ϕ_- with $\phi_+ > \phi_-$. Then, we define p_+ and p_- so that we have

$$\begin{aligned} P[q = q_+ | \phi = \phi_+] &= p_+ \\ P[q = q_- | \phi = \phi_+] &= 1 - p_+ \\ P[q = q_+ | \phi = \phi_-] &= p_- \\ P[q = q_- | \phi = \phi_-] &= 1 - p_-. \end{aligned} \tag{3.2}$$

Since the principal is unable to observe and enforce ϕ, it will instead provide an incentive to the agent in the form of a differentiated payment - a function of the project outcome. In this case, since there are only two possible outcomes q_+ and q_-, there will be two payments ψ_+ and ψ_- with $\psi_+ > \psi_-$. Thus, the principal will pay the agent ψ_+ if the outcome is q_+, and a smaller amount ψ_- if the outcome is q_-.

This situation can be modelled as a game:

1. **Players**: the *agent* p_s and the *principal* p_c.

[2] A nondecreasing, concave utility function captures elementary intuitions regarding the relative value of money. In particular, it implies that higher payments bring higher utility but with diminishing returns.

2. **Types**: p_s will have a reservation utility $U_r \in \mathbb{R}_{\geq 0}$.

3. **Actions and Strategies**: The agent can select its level of effort, and $\mathcal{A}_s = \{\phi_+, \phi_-\}$. The principal can select the payment it gives to the agent, and $\mathcal{A}_c = \{\psi_+, \psi_-\}$. In this case, the strategy for both the principal and the agent is just choosing an element in their action sets.

4. **Outcome**: There are only two possible outcomes, and $\mathcal{O} = \{q_+, q_-\}$. In this case, the outcome function is implicitly defined by (3.2).

5. **Utility Function**: U_s and U_c as defined before.

Let U_c^+ be the expected utility for the principal if $\phi = \phi_+$, and U_c^- the expected utility for the principal if $\phi = \phi_-$. Then, we have that

$$\begin{aligned} U_c^+ &= p_+ U_c(q_+, \psi_+) + (1 - p_+) U_c(q_-, \psi_-) \\ U_c^- &= p_- U_c(q_+, \psi_+) + (1 - p_-) U_c(q_-, \psi_-). \end{aligned}$$

In the same way, the expected utilities for the agent as a function of its effort and payment are

$$\begin{aligned} U_s^+ &= p_+ U_s(\psi_+, \phi_+) + (1 - p_+) U_s(\psi_-, \phi_+) \\ U_s^- &= p_- U_s(\psi_+, \phi_-) + (1 - p_-) U_s(\psi_-, \phi_-). \end{aligned}$$

The incentives problem for the principal is to calculate ψ_+ and ψ_- to maximise its expected utility U_c^+, while at the same time making $\phi = \phi_+$ a dominant strategy for the agent[3]. In other words, we seek to ensure that the agent obtains a higher utility by exerting ϕ_+ than by exerting ϕ_- (we call this the *incentive compatibility* constraint), and that the agent utility is high enough to compete with the labour market (the *rationality* constraint). The principal-agent problem is thus formulated in terms of the following optimisation problem.

$$\begin{aligned} &\text{Maximise:} \quad U_c^+ \\ &\text{Subject to:} \quad U_s^+ \geq U_r \quad \text{(rationality)} \\ &\text{And:} \quad U_s^+ \geq U_s^- \quad \text{(incentive compatibility).} \end{aligned} \tag{3.3}$$

We solve this problem using standard Kuhn-Tucker first order conditions derived from the Lagrangian

$$\begin{aligned} \mathcal{L}_A(\psi_+, \psi_-) = \quad & p_+(f_c(q_+) - \psi_+) + (1 - p_+)(f_c(q_-) - \psi_-) \\ & + \lambda(p_+ f_s \psi_+ + (1 - p_+) f_s \psi_- - \phi_+ - U_r) \\ & + \mu((p_+ - p_-)(f_s \psi_+ - f_s \psi_-) - \phi_+ + \phi_-). \end{aligned} \tag{3.4}$$

[3] We shall not consider optimising U_c^-.

The critical point condition $\nabla\mathcal{L} = 0$ yields the following first order conditions:

$$\begin{aligned}
(\lambda p_+ + \mu(p_+ - p_-)) \left.\frac{\partial f_s}{\partial \psi}\right|_{\psi_+} &= p_+ \\
((1 - \lambda p_+) - \mu(p_+ - p_-)) \left.\frac{\partial f_s}{\partial \psi}\right|_{\psi_-} &= (1 - p_+) \\
p_+ f_s(\psi)|_{\psi_+} + (1 - p_+) f_s(\psi)|_{\psi_-} &= \phi_+ + U_r \\
f_s(\psi)|_{\psi_+} - f_s(\psi)|_{\psi_-} &= \frac{\phi_+ - \phi_-}{p_+ - p_-}.
\end{aligned}$$

These equations can then be solved for ψ_- and ψ_+, the expression for which will of course be dependent on f_s.

We present a particular kind of principal-agent model and its novel application to overlay services provided over the best-effort Internet in Chapter 6.

3.4.2 Auctions and Mechanism Design

Consider a situation where a peer n_i (the *auctioneer*) offers a single item for sale among a set $\mathsf{P} = \{\mathsf{p}_j\}$ of strategic agents (the other peers). Each p_j has a value $v_j \in \mathbb{R}_{\geq 0}$ that represents how much it values the item being auctioned - how much it is willing to pay to obtain it. Therefore, if p_j wins the item and it is required to pay a price ν for it, the net utility that it will experience is $U_j = v_j - \nu$; if some other agent wins the item, $U_j = 0$.

The simplest way in which that the auctioneer could resolve the auction would be to allocate the item to that agent who values it the most: n_w such that $w = \arg\max_j v_j$. The problem is that n_i cannot know the true values v_j, since they are private to each p_j. Since the bidders can report any value that they choose, the challenge is to provide them with an incentive to truthfully reveal their v_j. We now show that this is achievable simply by correctly choosing the price ν that the winning peer will pay for the item.

The *Vickrey auction* provides a mechanism in which it is a dominant strategy equilibrium for bidders to report their valuations truthfully, and that succeeds in assigning the auctioned items to those peers that value them the most (it is *efficient*). We now analyse it in the single-item case.

3.4.2.1 The Single-Item Vickrey Auction

The single-item, sealed-bid, second-price auction is defined as the following game:

- **Players**: We assume that an *auctioneer* peer n_i will auction a single item to the rest of the peers. Thus, we have that $\mathsf{P} = \mathsf{N} \setminus n_i$. Of course, $|\mathsf{P}| = (|\mathsf{N}| - 1)$ and we assume that $|\mathsf{P}| \geq 2$.

- **Types**: Each bidding peer $n_j : j \neq i$ will be a player p_j of the game, with a private *value* v_j for the item that is being auctioned. Therefore, in this case, $\zeta_j = v_j$, $\mathcal{Z}_i = \mathbb{R}_{\geq 0}$ and $\mathcal{Z} = \mathbb{R}_{\geq 0}^{|\mathsf{P}|}$.

- **Actions and Strategies**: Since the only action available to a peer corresponds to sending a bid, its only strategy is choosing the amount to bid. Accordingly, the strategy space of each peer is $\mathcal{S}_i = \mathbb{R}_{\geq 0}$, and the strategy space of the game is $\mathcal{S} = \mathcal{S}_1 \times \mathcal{S}_2 \times \ldots \times \mathcal{S}_{|\mathsf{P}|} = \mathbb{R}_{\geq 0}^{|\mathsf{P}|}$. The strategy s_i will be equal to the amount that a peer bids, and the strategy profile of a game is just a vector $s \in \mathbb{R}_{\geq 0}^{|\mathsf{P}|}$.

- **Outcome**: The outcome of the game consists of the identity of the peer that wins the item, $\mathsf{p}_w \in \mathsf{P}$, and the amount ν paid for the item. The outcome function is defined to be $\mathsf{o}(s) = (\mathsf{o}_w(s), \mathsf{o}_\nu(s))$ where $\mathsf{o}_w(s) = \{\mathsf{p}_w : w = \arg\max_j s_j\}$ and $\mathsf{o}_\nu(s) = \{\max_{j \neq w} s_j\}$. Thus, the bidder with the

highest bid wins the item, and pays the highest non-winning bid. If we denote the second highest bid as $\hat{\nu}$, we have that $\nu = \hat{\nu}$.

- **Utility Function**: The winning bidder p_w has a utility of $U_j = v_j - \hat{\nu}$, all the other peers have a utility of 0.

The Vickrey auction has the property that any given peer can never increase its utility by bidding any value different from its true valuation v_j. This means that truthfully revealing their private values is a *dominant strategy* for all peers. Formally,

Theorem 3.2. *Let p_j be any given bidder and s_j its bid value. Let $U_j^* = U_j|_{s_j=v_j}$ be the utility that the bidder obtains if it truthfully reveals its value, and $U_j^s = U_j^s|_{s_j \neq v_j}$ the utility that it obtains if it strategically reports $s_j \neq v_j$. Then, $U_j^* \geq U_j^s$.*

Proof. There are two outcomes for the auction as far as p_j is concerned: it can either win the auction or lose it. We analyse each one in turn.

- Assume that p_j would **win** the auction by reporting $s_j = v_j$, and that the price to be paid in that case is $\nu = \hat{\nu}$. In this case, $U_j^* = v_j - \hat{\nu} \geq 0$, because $\hat{\nu}$ is the *second* highest bid after $s_j = v_j$. Now, we consider the two possibilities regarding an strategically defined s_j:
  - **Case 1** ($s_j \leq \hat{\nu}$): In this case, p_j would lose the auction. Since in this case $U_j^s = 0$, we have that $U_j^* > U_j^s$. Of course, we assume that if $s_j = \hat{\nu}$ there will be some arbitrary tie-breaking procedure. In this case, we assume that this process goes against p_j and it loses the auction. The specifics of this are irrelevant for the proof.
  - **Case 2** ($s_j > \hat{\nu}$): In this case, p_j would still win the auction and pay the same price, since $\hat{\nu}$ is unaffected. In this case, we have that $U_j^* = U_j^s$.
- Assume that p_j would **lose** the auction by reporting $s_j = v_j$. In this case, $U_j^* = 0$, and some other peer wins with bid $\hat{\nu}$. Again, we consider the two possibilities regarding s_j:
  - **Case 3** ($s_j \leq \hat{\nu}$): In this case, p_j would lose the auction anyway, and $U_j^s = 0$. So, we have that $U_j^* = U_j^s$.
  - **Case 4** ($s_j > \hat{\nu}$): In this case, p_j would win the auction and pay $\hat{\nu}$ as price. However, this price is too high for p_j, because $v_j < \hat{\nu}$ - that is the reason p_j would have lost the auction had it bid truthfully. Thus $U_j^s = v_j - \hat{\nu} \leq 0$, and $U_j^* > U_j^s$.

It follows from the previous discussion that $U_j^* \geq U_j^s$ irrespective of v_j and $\hat{\nu}$. □

The Vickrey auction is a very elegant solution to a seemingly very difficult problem: it can reliably compute a function ($\arg\max$) over a set of numbers, each of which is being held by a strategic agent that has no a-priori reason to reveal it truthfully. The Vickrey auction is able to solve a welfare-maximisation problem even if the inputs for the problem are private to each strategic agent, which in turn acts only on the basis of maximising its own utility.

3.4.2.2 The Vickrey-Clarke-Groves Mechanism

The easiest way to generalise the single-item Vickrey auction to multiple items is through *mechanism design*. Mechanism Design allows a designer to specify the rules of a game in order to achieve, as an equilibrium result, a specific social outcome, even if the many agents which will participate in the game are self-interested and act in a strategic manner. This is achieved by defining a set of incentives for the agents, such that behaviour resulting in the desired social outcome is preferred to other courses of action.

These agent incentives usually include *monetary payments* that are functions of the behaviour of the agents. Although there are ways of obviating the need for money in mechanism design ([160, 232] and [242] Chapter 10), there is no need for such techniques in this case. We will use the contribution transfer protocol primitives explained in Chapter 4 to achieve payments. Instead of exploring the generic issues surrounding mechanism design (see [241, 174, 169, 220], [227] Chapter 6 or [242] Chapter 9) we focus on the *Vickrey-Clarke-Groves* (VCG) mechanism and its relationship to multi-item Vickrey auctions.

The VCG mechanism, when defined as a game, has the following characteristics (see Table 3.2 for variable definitions):

- **Players**: We assume that a peer n_i is distinguished as a mechanism *designer* that wishes to implement an optimal social outcome among a set of players P (as before, $\mathsf{P} = \mathsf{N} \setminus n_i$).

- **Types**: Each peer $n_j : j \neq i$ will be a player p_j of the game, with a private type ζ_j that defines how much value a particular social outcome $\mathsf{k} \in \mathcal{K}$ holds for the peer. This is done through *valuation function* $v_j(\mathsf{k})$. Thus, different players value social outcomes differently, according to their type and their associated valuation function. Since $v_j(\mathsf{k})$ is private, it can be used to model not only endogenous peer characteristics, but information about the social environment which is only known to that player.

- **Actions and Strategies**: We deal only with the most basic kind of VCG mechanism, where each peer p_j sends to the designer a *message* that states its preferences regarding the social outcome, and then designer the announces the social outcome and payments to be applied to each player based on the messages it receives. Formally, this means that each peer will either supply its true valuation function $v_j(\mathsf{k})$ or an strategically defined one $s_j(\mathsf{k})$.

- **Outcome**: When the designer receives a vector $s(\mathsf{k}) = (s_1, s_2, \ldots, s_\mathsf{P})$ of player messages (each one a function of k), it applies an *outcome function* $\mathsf{o}(s(\mathsf{k})) = (\mathsf{o}_k(s(\mathsf{k})), \mathsf{o}_h(s(\mathsf{k})))$ to select what social outcome $\mathsf{k}^* = \mathsf{o}_k(s(\mathsf{k}))$, will be selected, and what payments $\mathsf{h}^* = \mathsf{o}_h(s(\mathsf{k}))$ will be applied to the players.

- **Optimal Social Choice Implementation**: The mechanism chooses the social outcome $\mathsf{k} \in \mathcal{K}$ that maximises the aggregate reported player value $\hat{s}(\mathsf{k}) = \sum_{\mathsf{p}_j} s_j(\mathsf{k})$ as an objective function. Therefore, $\mathsf{k}^* = \arg\max_{\mathsf{k} \in \mathcal{K}} \hat{s}(\mathsf{k})$, and the mechanism designer maximises the total aggregate value, according to the valuation functions $s_j(\mathsf{k})$ supplied by the peers.

- **Manipulation-resistant payments**: The VCG payments h_j that each peer receives can be expressed as

$$\mathsf{h}_j(s(\mathsf{k})) = \mathsf{c}_j(s_{-j}) - \sum_{k \neq j} s_k(\mathsf{k}), \tag{3.5}$$

  where the c_j are arbitrary functions of s_{-j} only. Thus, since $k \neq j$ in the summation above, $\mathsf{h}_j : \mathcal{S}_{-j} \mapsto \mathbb{R}_{\geq 0}$ and the payment that an agent p_j receives is never a function of its own strategy s_j. Hence, a player can not increase its payment by selecting an advantageous s_j. With the aforementioned payments, the utility of each peer becomes

$$U_j(\mathsf{k}) = v_j(\mathsf{k}) + \sum_{k \neq j} s_k(\mathsf{k}) - \mathsf{c}_j(s_{-j}). \tag{3.6}$$

- **Utility Functions**: Each p_j has a utility of $U_j(\mathsf{k}) = v_j(\mathsf{k}) - \mathsf{h}_j(\mathsf{k})$, where h_j are the payments that p_j will have to pay.

k	Social outcome in of an economic mechanism
$\mathcal{K}$	Set of all possible social outcomes
$v_j(\mathsf{k})$	Valuation of a social outcome k by agent p_j
$s_j(\mathsf{k})$	Valuation of a social outcome k that agent p_j reports to the mechanism designer
$\hat{s}(\mathsf{k})$	Aggregate valuation of a social outcome k to all agents, according to their reported valuation functions
h_j	Payment to agent p_j
$U_j(\mathsf{k})$	Utility of agent p_j for social outcome k
$\mathsf{c}_j(s_{-j})$	Cost that can be applied to agent p_j as a function of the strategies of all peers except itself

Table 3.2: Mechanism design notation

- **Incentive Compatibility**: The VCG mechanism is *incentive compatible*: truthfully revealing its valuation function $v_j(\mathsf{k})$ is a dominant strategy for all players - there is no incentive for any p_j to unilaterally deviate from revealing its true valuation. To see why, consider a given p_j with valuation $v_j(\mathsf{k})$, reported valuation $s_j(\mathsf{k})$ and a fixed but arbitrary strategy profile s_{-j} for the rest of the peers. Let $\mathsf{k}^* = \mathsf{o}_k(\{v_j, s_{-j}\})$ and $U_j^* = U_j|_{s_j=v_j}$ be the social choice and the utility of p_j if it truthfully reports its value, and $\mathsf{k}^s = \mathsf{o}_k(s_j, s_{-j})$ and $U_j^s = U_j|_{s_j \neq v_j}$ those obtained if it strategically reports a different valuation function $s_j(\mathsf{k})$. We see that, by (3.6),

$$U_j^* + \mathsf{c}_j(s_{-j}) = v_j(\mathsf{k}^*) + \sum_{k \neq j} s_k(\mathsf{k}^*) := \hat{s}(\mathsf{k}^*)$$
$$U_j^s + \mathsf{c}_j(s_{-j}) = v_j(\mathsf{k}^s) + \sum_{k \neq j} s_k(\mathsf{k}^s) := \hat{s}(\mathsf{k}^s).$$

  However, we know that k^* was calculated by the mechanism designer so that $\mathsf{k}^* = \arg\max_{\mathsf{k} \in \mathcal{K}} \hat{s}(\mathsf{k})$, which means that $\hat{s}(\mathsf{k}^*) \geq \hat{s}(\mathsf{k}^s)$. This means that $U_j^* + \mathsf{c}_j(s_{-j}) \geq U_j^s + \mathsf{c}_j(s_{-j})$, which in turn implies that $U_j^* \geq U_j^s$.

- **Incentives to players**: If we assume that the p_j truthfully reveal their valuations $v_j(\mathsf{k})$, their utilities become $U_j(\mathsf{k}) = \sum_k v_k(\mathsf{k}) - \mathsf{c}_j(s_{-j})$. This means that, in the process of maximising its own utility, each agent will choose the outcome k^* that maximises the social utility (the sum of all peer values).

Until now, the $\mathsf{c}_j(s_{-j})$ have been essentially arbitrary, with the only requirement of not depending on s_j. However, by an appropriate choice of $\mathsf{c}_j(s_{-j})$, the VCG mechanism can gain additional benefits. One possible selection that makes the VCG mechanism *individually rational* and *positive transfer free* is the *Clarke Pivot Rule*, by which

$$\mathsf{c}_j(s_{-j}) = \max_{\mathsf{k} \in \mathcal{K}} \sum_{k \neq j} s_k(\mathsf{k}).$$

Informally, the Clarke pivot rule assigns to $\mathsf{c}_j(s_{-j})$ the aggregate reported value *had* p_j *not participated in the mechanism at all*. In this case, if we assume truthful reporting of values, we have that, for the player payments,

$$\mathsf{h}_j(\mathsf{k}^*) = \max_{\mathsf{k} \in \mathcal{K}} \sum_{k \neq j} v_k(\mathsf{k}) - \sum_{k \neq j} v_k(\mathsf{k}^*).$$

Thus, each player pays an amount equal to the value degradation that it causes to the rest of the players: the difference between the total value that the system would have obtained if it had not participated, and

the total value that the rest of the system achieves if it does participate. From this, we see that the utility that a given player obtains from the mechanism (assuming each player truthfully reveals its valuation and optimises its own utility) is

$$U_j = \max_{\mathsf{k} \in \mathcal{K}} \sum_k v_k(\mathsf{k}) - \max_{\mathsf{k} \in \mathcal{K}} \sum_{k \neq j} v_j(\mathsf{k}), \tag{3.7}$$

which means that the total utility that a player obtains is just the extra utility that it brings into the game with its participation - the difference between the total utility of the system with and without its presence.

The VCG mechanism with Clarke pivoting (VCG-C) has the following two additional important properties:

- **Individual rationality**: The players in a VCG-C mechanism will always get non-negative utility.

  Theorem 3.3. *If* $v_j(\mathsf{k}^*) \geq 0$ *for every* p_j *and* k^*, $U_j \geq 0$.

  Proof. Let $\mathsf{k}^* \in \mathcal{K}$ be the optimal social choice if p_j participates in the mechanism, and $\mathsf{k}^*_{-j} \in \mathcal{K}$ the optimal social choice if it does not. In addition, let $\mathsf{V}(\mathsf{k}) = \sum_k v_j(\mathsf{k})$ be the total value of the mechanism if p_j participates. Then, we have that

  $$U_j = \mathsf{V}(\mathsf{k}^*) - \mathsf{V}(\mathsf{k}^*_{-j}) + v_j(\mathsf{k}^*_{-j}).$$

  However, since $v_j(\mathsf{k}^*_{-j}) \geq 0$, and because k^* maximises $\mathsf{V}(\mathsf{k})$, we have that

  $$U_j \geq \mathsf{V}(\mathsf{k}^*) - \mathsf{V}(\mathsf{k}^*_{-j}) \geq 0,$$

  and it follows that $U_j \geq 0$. □

- **No positive transfers**: The players never receive money from the designer.

  Theorem 3.4. *If there is truthful reporting,* $\mathsf{h}_j(\mathsf{k}) \geq 0$.

  Proof. Let $\mathsf{V}_{-j}(\mathsf{k}) = \sum_{k \neq j} v_j(\mathsf{k})$ be the total value of the mechanism if p_j does not participate, and $\mathsf{k}^*_{-j} \in \mathcal{K}$ the optimal social choice in this same case. Again, let $\mathsf{k}^* \in \mathcal{K}$ be the optimal social choice if p_j participates in the mechanism. We can see that

  $$\mathsf{h}_j(\mathsf{k}^*) = \mathsf{V}_{-j}(\mathsf{k}^*_{-j}) - \mathsf{V}_{-j}(\mathsf{k}^*) \geq 0,$$

  because k^*_{-j} was chosen to optimise $\mathsf{V}_{-j}(\mathsf{k})$. □

3.4.2.3 The Multi-Item Vickrey Auction

By using the VCG-C formalism, the properties of the multiple-item Vickrey auction can be easily explored. Although analysing the general case where there is a distinct valuation for each item (and indeed for each combination of items) is simple once the VCG-C mechanism has been presented, we focus on the case where the items offered are identical, so that the utility that a player gets from participating in the auction is only a function of the number of items it wins.

A seller peer n_i wishes to sell m identical items to the other peers, so that $\mathsf{P} = \mathsf{N} \setminus n_i$ and the social outcome k is the allocation of items to bidders. Formally, k is a vector with one component for each $\mathsf{p}_j \in \mathsf{P}$, where k_j indicates how many items were assigned to p_j. Each bidder provides the auctioneer with a valuation function $v_j(\mathsf{k}_i)$ indicating its benefit in obtaining k_i items, for $0 \leq \mathsf{k}_i \leq b_j$, where b_j

is the maximum number of items that p_j is interested in. The auctioneer then then calculates the value-maximising allocation $\mathsf{k}^* = \arg\max_{\mathsf{k}} \sum_{\mathsf{p}_j} v_j(\mathsf{k}_j)$ subject to the feasibility constraint $\sum_j \mathsf{k}_j \leq m$. The prices that each bidder will pay are calculated are as indicated by (3.5), and the utilities as indicated by (3.7). We analyse this particular kind of multi-item Vickrey auction in much greater depth in Chapter 5, in the context of a novel auction-based swarming protocol for QoS overlays.

4

A Sybilproof Indirect Reciprocity Mechanism

Although direct reciprocity (*Tit-for-Tat*) contribution systems have been successful in reducing freeloading in peer-to-peer overlays, it has been shown that, unless the contribution network is dense, they tend to be slow or fail [198]. On the other hand, current indirect reciprocity mechanisms based on reputation systems tend to be susceptible to *sybil attacks*, *peer slander* and *whitewashing* (See Section 3.2.1).

In this chapter, we present *PledgeRoute*, an accounting mechanism for peer contributions that is based on *social capital* (see Appendix A.3 and [56, 127, 86] for the social sciences definitions of this term). This mechanism allows peers to contribute resources to one set of peers and use these contributions to obtain services from a different set of peers, at a different time. *PledgeRoute* is completely decentralised, can be implemented in both structured and unstructured peer-to-peer systems, and is resistant to the three kinds of attacks mentioned above. Our focus on the expression of social capital as a set of transitive obligation chains stems from the analysis of complementary currencies, where it has been used as a medium of exchange to support economic transactions [273, 206, 36, 287].

To achieve this, we model contribution transitivity as a routing problem in the *contribution network* of the peer-to-peer overlay, and we present arguments for the routing behaviour and the sybilproofness of our contribution transfer procedures on this basis. Additionally, we present mechanisms for the seeding of the contribution network, and a combination of incentive mechanisms and reciprocation policies that motivate peers to adhere to the protocol and maximise their service contributions to the overlay. We elaborate on these contributions in the following sections.

4.1 Freeloading, Reciprocity and Contribution Accounting

Peer-to-peer overlays coordinate the contributions of large numbers of independent peers to form scalable, decentralised, self-organising content delivery systems. However, as explored in Section 2.1, they are susceptible to the *free-riding problem* ([27, 254]): it is individually rational for each peer to contribute as little as possible, while at the same time consuming the contributions of other peers.

If resource contribution is framed as an iterated *Prisoner's Dilemma*, the expectation for reciprocity in future interactions (including retaliation) can be a strong incentive for cooperation [39]. Reciprocity

then emerges as a viable technique for the control of freeloading. We follow the standard nomenclature for **direct** and **indirect** reciprocity (see Chapter 2 and Appendix A.4).

Many resource allocation techniques based on direct reciprocity have been proposed (see, for instance, [84, 185, 97]). These, although easily implementable on the peers, can be limited by the high churn rates typical of peer-to-peer overlays. Any given peer may be interacting essentially with strangers, and thus, direct reciprocity mechanisms will tend to have slow convergence, if they converge at all [198]. In practice, these mechanisms are only robust for systems with long-lived relationships where the opportunity for mutual reciprocation is high.

Direct reciprocation is further complicated if the subset of peers that provide the services that a peer seeks is not the same one that is interested on the services that it provides. This supply-demand mismatch can also happen in time, since peers are usually only interested in particular services at particular times - which means peers can find that their past contributions could be useless when it comes to obtaining services in the present. Given that peers that do not consume a stream but still act as bandwidth multipliers (a recent example of this would be [319, 318]) can potentially play an important role in increasing the total system capacity in streaming QoS overlays, we propose a scheme that will allow them to take advantage of their past contributions from the peer and use them for stream consumption when needed, even if the set of peers that received past contributions and the set of peers from which service is requested are disjoint.

In these circumstances, a reciprocity system that allows contributions given to a peer to be "repaid" by other peers at different times in the future may be the only way to foster cooperation between strategic peers seeking private utility maximisation.

Most indirect reciprocity schemes proposed in the literature (see, for instance [185, 153, 97] and Chapter 2.7) are based on *reputation systems*, and differ mainly in the way the reputation scores are calculated and propagated. In general, however, they assign ratings to peers according to their past behaviour, and communicate them through the overlay itself. This makes them vulnerable to exploitation, due to the near-zero cost of creating new identities. Peers might create arbitrary contributions between fictitious peers (the *sybil attack* [106]), lie regarding their contributions or the contributions of other peers (the *slander* attack), or discard identities that have been labelled as malicious and penalised (the *whitewashing* attack).

Other indirect reciprocity schemes rely on implementing a currency-based economy that is resistant to forgery and double-spending (see, for instance, [328] and [132], and Section 2.5). A common problem of these schemes is the minting of currency and the trust anchoring that it implies, usually necessitating either a public key infrastructure, a web of trust, or threshold cryptography techniques. Furthermore, these systems usually make use of auctions (See Section 2.5.5), which usually place on the peer the nontrivial burden of estimating the value of the services that other peers offer.

4.1.1 PledgeRoute

In this chapter we propose *PledgeRoute*, a contribution accounting system that provides indirect reciprocity by allowing peers to contribute to a set of peers and transfer these contributions for use with other peers in a direct reciprocity basis. Thus, we enable indirect reciprocity by decomposing it in two stages: *contribution transfer* and *direct reciprocation*.

Contribution transfer operates by decomposing an end-to-end transfer operation into a sequence of pairwise contribution transfers between peers that have interacted in the past. Any peer n_i can perform contributions towards any other given peer n_j, which will be converted to an abstract measure of contribution. Then, this n_i can then request to have its contribution transferred from n_j to a different peer n_k, from which it can now request any service. Instead of relying on centralised markets, we propose an accounting system that not only keeps track of peer contributions, but also allows them to be trans-

ferred between peers following contribution chains. Peers can contribute to one set of peers and transfer these contributions to another set of peers from which services are required, implementing a distributed exchange economy.

By casting contribution transitivity as a routing problem, we can use distributed routing algorithms to achieve our objective in a completely decentralised fashion. To achieve this, we propose a Dijkstra-inspired generalised routing algorithm to route contributions through the network by means of *wealth-preserving* transactions. In order to make the system sybilproof [71], our algorithm for contribution transfer relies on finding the end-to-end transfer paths that have maximum bottleneck contributions, which have sybilproofness properties similar to those of maximum flow. Peers use self-certifying identifiers [223] that their neighbours can verify with public keys that are exchanged when the peers initially come into contact (all communication between peers is digitally signed). Thus, since contributions are always bound to an identity, peers gain nothing from having multiple identities: their contributions will be simply split among them. To complement our routing algorithm, we present simple cryptographic techniques and protocol operations that provide resistance to slander and whitewashing attacks.

To avoid the execution of costly routing computations over the entire peer-to-peer overlay topology, we propose a probabilistic topology sampling algorithm based on a truncated, self-avoiding random walk that samples preferentially those paths capable of yielding high-valued contribution transfers. Using information recovered from messages forwarded along these random walks, each peer constructs a local contribution network model of the peers that have unreciprocated contributions to one another and that can be used to find paths with high contribution transfer potential. Contribution transfer is achieved using a soft-state reservation protocol.

Since the contribution transfer operation only preserves the net contributions of each peer, and not its absolute contributions and obligations, there is a danger of draining the contribution network of social capital (we discuss this problem in Section 4.5). This means that, although no peer ends up "worse off", the capacity of the system to transfer contributions from one peer to another will be reduced. To counter this, we propose a technique to seed the contribution network based on the creation of a credit tree rooted on each peer, and its use for the creation of balanced contribution cycles.

To achieve stable cooperation in the reciprocation phase in the presence of strategic peers, we propose an incentive mechanism based on the modification of the contribution values of each peer.

4.2 Definitions

In this chapter we will be mainly concerned with the *contribution network*: the graph-theoretic representation of the contributions that have been given and received in the peer-to-peer overlay. We will model the contribution network as link-weighted, directed graph $\mathcal{G}$ using the nomenclature of Section 3.1, where the link weights $w_{ij} \geq 0$ associated with the links l_{ij} correspond to the magnitude of the contributions from n_i to n_j. In practice, w_{ij} is just an account associated with n_i that is maintained in n_j. Again following Section 3.1, we denote a *simple path* in $\mathcal{G}$ from n_i to n_j as P_{ij} (we discuss the weighing of paths and links in the context of our routing algebra in Sections 4.4 and 4.7).

We shall denote $w_{-}(n_i)$, the weighted outdegree of n_i (the total unpaid contributions that n_i has given to its neighbours) as its *social capital*. Conversely, we shall denote its weighted indegree $w_{+}(n_i)$ (the total unpaid contributions that n_i has received from its neighbours) as its *pledged resources*. We call the net contribution of n_i to the system its *wealth*, and we define it as $W(n_i) = w_{-}(n_i) - w_{+}(n_i)$.

We shall be interested in modifying the topology of $\mathcal{G}$ (usually by creating new links) while maintaining the net contributions of each peer invariant. To perform such *wealth preserving* transformations, we begin by defining a closed cycle $C \in \mathcal{G}$ on the contribution network, and we assign an orientation to it. If we need to create new links in $\mathcal{G}$, we consider them to have zero weight ($w_{ij} = 0$).

Having defined C, we traverse its links following its orientation, adding an amount τ to the contributions associated with each link that is traversed in its designated direction, and subtracting an amount τ from the contributions associated with each link that is traversed opposite to its designated direction (transformations requiring negative link weights are considered invalid). Each peer in C adds the same amount to its indegree and its outdegree, keeping its wealth invariant. After the transformation, links with zero weight are removed from $\mathcal{G}$.

Every peer n_i sets υ_{ij}, the maximum amount of unreciprocated contributions that it will give to a neighbour n_j in $\mathcal{G}$. We will call υ_{ij} the *safe credit margin* of n_j as assessed by n_i. In order to calculate υ_{ij}, each peer n_i maintains $\hat{w}_{ij}$, the amount of contributions that n_j has given to n_i as response to reciprocative requests (see Section 4.3). As peers successfully reciprocate for a number of interactions they can be more reliant on the continued trustworthiness of their neighbours, and υ_{ij} will increase with $\hat{w}_{ij}$. Finally, each peer enforces ξ_i, the maximum credit it is willing to support. We defer to Section 4.5 the analysis of the relation between w_{ij}, $\hat{w}_{ij}$ and υ_{ij}.

4.3 Basic System Concepts

In *PledgeRoute*, the contribution network $\mathcal{G}$ represents the set of previously rendered services, which carries a reciprocity implication and is equivalent to the notion of *social capital* in the social sciences [86]. Thus, we have a web of commitments, where social capital (unreciprocated past contributions) can be freely converted into services (for instance, streaming traffic).

To avoid single points of failure or the compulsory presence of trusted third parties, we require no centralised identity management system. Instead, peers use self-certifying identifiers that are exchanged when they initially come into contact. These can be used as public keys, to verify digital signatures on the messages sent by their neighbours, as all communication between peers is digitally signed. An additional benefit of this decoupling of the incentives mechanism from the identity management system is that the protocol can be easily used in cases where anonymous or pseudonymous participation is desired. This, of course, does not mean that we *require* user anonymity or pseudonymity - the user can be authenticated in any relevant way in order to enable access to content or functionality, without this affecting the operation of the system in any way.

There are two distinct interaction policies in our system: *altruistic* and *reciprocating* (these will be explained shortly). A peer chooses one of these two when requesting a service, and the behaviour of the server peer depends on this choice.

When a peer requests a service using the altruistic policy, it does not offer any kind of "payment" for it, relying instead on the altruism of the serving peer. The serving peer, however, will only grant the service on a best-effort basis: its service quality will be reduced as required, giving priority to reciprocation-based interactions.

However, altruistic services are not "free": the serving peer n_j will expect reciprocation at some future time from the requesting peer n_i (this is equivalent to the *reciprocal altruism* of [299]). This expectation takes shape as an increase in w_{ji}, and models the reciprocative obligation that the recipient has contracted towards the serving peer.

Although any request using the altruistic policy receives essentially random service quality, this policy is of fundamental importance as it helps the system *bootstrap*: when new peers arrive in the system, they have no previous contributions to other peers in the overlay, and they can only obtain services through altruism (at least until they have had the opportunity to provide services themselves).

When a peer n_i requests a service from a server peer n_j using the reciprocating policy, it will offer a payment τ to cover it that will be deducted from the account w_{ij} in n_j (the previous contributions that n_i has made to n_j, including contributions that n_i has transferred to n_j from other peers). If τ is sufficient

to cover for the request, n_j will give it prioritised service and subtract τ from w_{ij}. If τ is insufficient or $w_{ij} < \tau$, the request will be interpreted as an altruistic interaction, and the request originator will be informed of this fact. This should be uncommon, as the originator is aware of its contributions to the server peer. We assume that the conversion between services and contribution units is the same for all peers, and universally known[1].

Clearly, there is an element of trust involved in this transaction: the value of the service that the client peer receives might not be equivalent to the amount τ by which its account is decreased. This problem is equivalent to the server n_j arbitrarily decreasing the client's account w_{ij} by an amount equal to the difference between τ and the true value of its service[2]. The reaction of the client to this is detailed in Section 4.6.

In summary, altruistic interactions **increase** the social capital in the network, while reciprocative interactions **decrease** it. We now explore the use of these accounts for contribution transfer.

4.4 Distributed Contribution Accounting

We now describe the protocol elements of *PledgeRoute*, focusing in those properties relevant to the issues detailed in Section 4.1. In particular, we shall not focus in the peculiarities of service pricing and valuation, instead choosing to complement other works such as PeerMart [161] and [82].

4.4.1 Contribution Topology Discovery

The first issue that we will address is the process that allows each peer to construct a local view of the contribution network $\mathcal{G}$ that can be used for the computation of contribution transfers. This local view of $\mathcal{G}$ from the standpoint of n_i will be called $\mathcal{G}_i$. The aim of our topology discovery algorithm is to allow peers to attain high levels of contribution transfer capability while keeping as few links of $\mathcal{G}$ in $\mathcal{G}_i$ as possible.

The contribution topology discovery process starts when a peer advertises the unpaid contributions that it has to each of its neighbours in $\mathcal{G}$ using *Pledge Announcement Messages* (PAMs). Each PAM is associated with a simple path in $\mathcal{G}$: a series of links terminating on the sending peer where no peer or link is repeated.

The path that the PAM follows is decided in a probabilistic fashion, with the probability of it being forwarded from a peer n_i to a given neighbour peer n_j made proportional to w_{ji}, the unpaid contributions that n_j has made to n_i (the pledged resources that n_i has towards n_j). Clearly, the PAM has greater probability of traversing paths where peers have made large contributions.

We model this using a **truncated, self-avoiding random walk** defined on the contribution network $\mathcal{G}$. Peers periodically generate PAMs which are sent in random walks following the links in $\mathcal{G}$. These random walks, however, are biased to preferentially traverse links that have high contribution values, yielding biased topology samples that favour subgraphs with high contribution transfer potential.

We focus on the transition probability associated with a single instance of the random walk starting in peer n_s. If we define $p_j^t(s)$ as the probability of a PAM starting at peer n_s visiting peer n_j at time t, the state transition probability for our random walk is given by

$$p_j^{t+1}(s) = \rho_{\text{stop}}\delta_{js} + (1-\rho_{\text{stop}}) \sum_{i \in \Pi^+(n_j)} r_{ij}^t p_i^t(s)\,, \tag{4.1}$$

where r_{ij}^t denotes the probability of a transition from state i to state j (the forwarding of the PAM from n_i to n_j) at time t, ρ_{stop} denotes the probability that the PAM walk is terminated, and δ_{js} is the

[1] We drop this assumption in Chapters 5 and 6, where we generalise *PledgeRoute* to a decentralised currency platform.

[2] The problem of the client refusing to pay the server after it has granted a service does not exist, as the server locally maintains the account with the contributions of the client.

Kronecker delta. It is apparent from (4.1) that the system will periodically regress to its initial state s, corresponding to peer n_s. This reflects the fact that the PAM will be removed from the network and returned to its originator n_s. Periodically, new PAMs will be generated by the originating peer n_s, each one of which which will start another topology discovery random walk.

In order to make (4.1) self-avoiding and biased towards high-contribution paths, we define for the transition probabilities that

$$r_{ij}^t = \begin{cases} \frac{w_{ji}}{\sum_{j \in \mathcal{U}(n_i,P)} w_{ji}} & \text{if } \mathcal{U}(n_i,P) \neq \emptyset \\ \delta_{js} & \text{otherwise.} \end{cases} \tag{4.2}$$

In (4.2), $\mathcal{U}(n_i, P)$ is the *unvisited neighbour set* of n_i for path P: $\mathcal{U}(n_i, P) = \mathit{\Pi}^-(n_i) \backslash P$. Informally, $\mathcal{U}(n_i, P)$ is the set of peers that have unreciprocated contributions towards n_i which have not forwarded the PAM yet, and P is the path in $\mathcal{G}$ that the PAM has taken from its creation and until time t.

By design, r_{ij}^t will favour transitions from n_i to peers n_j with large w_{ji} values, avoiding previously visited peers, and terminating the walk if $\mathcal{U}(n_i, P) = \emptyset$. In practice, this random walk is implemented by including the sequence of visited peers on each PAM. Every peer will extract the neighbours that have not been visited by the PAM and then pick one randomly, the distribution being weighted by the contributions that each eligible neighbour has given to n_i (the value of the w_{ji} account). So, peers that can support higher contribution transfers are preferentially selected.

After a given peer (say n_j) is selected for forwarding, n_i will insert in the PAM a timestamp, a nonce, the self-certifying identifier of n_j, the maximum contribution value w_{ji} available for contribution transfer, and will digitally sign of all these elements. When n_j receives the PAM, it can check that it was indeed originated by n_i, that it is actually directed to itself, that the contribution value w_{ji} reported by n_i is correct, and that it is not a replayed message. This last check is done by ignoring all expired PAMs (those with timestamps older than a given timeout value t_{old}), and keeping track of all the nonces corresponding to messages that have not expired.

As the PAM propagates according to (4.1) and (4.2), each peer will include its own identifier, the contribution value that it has for the next peer in the chain and the rest of the elements detailed above. Given that every peer can trivially confirm or deny the value w_{ji} claimed by an upstream PAM peer, every PAM message is audited at each step of the random walk and it can be considered truthful.

After processing the PAM, every peer on the random walk will be able to update its local contribution subgraph $\mathcal{G}_i$ with the w_{ij} values corresponding to the links that the PAM had previously traversed. Since links are traversed in "inverse" order, each peer will update its local view of the contribution network with a directed contribution path rooted on itself and terminating on the originator of the PAM. When a given peer decides to terminate the PAM walk, it will forward it back to its originating peer (the reason for this is explored in Section 4.6).

4.4.2 Contribution Transfer Protocol

Once a subgraph $\mathcal{G}_i$ of the unreciprocated contribution topology $\mathcal{G}$ has been discovered by incorporating the information of a number of PAMs, the next step is to use it to perform contribution transfers. This is outlined in Figure 4.1.

Contribution transfer events can be decomposed as consecutive pairwise transactions. Thus, past contributions are only needed between adjacent peers in the contribution chain. However, since $\mathcal{G}_i$ may not perfectly reflect $\mathcal{G}$, we design the transfer protocol to be resistant to errors in $\mathcal{G}_i$.

If n_i requests to transfer up to r_i contribution units to a "remote" peer n_k, the next-hop peer n_j might be unable to immediately commit, as this is contingent on changing conditions in the downstream contribution chain. Instead, n_j can calculate a contribution value that itself is willing to reserve for the

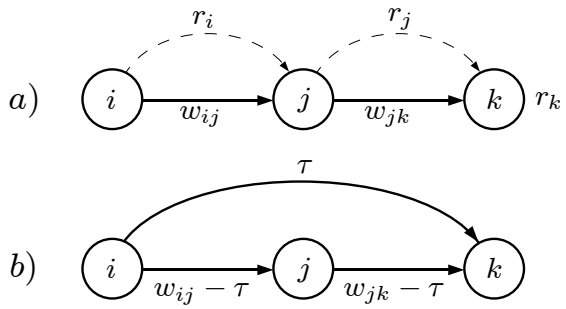


Figure 4.1: Contribution paths and transfers

transaction (r_j in Figure 4.1) and forward this information on. If every peer in the contribution chain does this, the final peer in the chain (in this case n_k, the peer that will actually provide the service) will have a complete view of the amounts that each peer is willing to commit (including itself), and will be able to compute an amount τ that complies with all these requirements (Fig. 4.1a)[3].

After τ is found, it is deducted from the accounts w_{ij} in the contribution path, and added to the account for the origin peer on the destination peer (w_{ik} in Figure 4.1b). Then, the final result of a contribution transfer of magnitude τ is to create a contribution link of τ from n_i to n_k through a wealth preserving operation, subtracting τ from every hop in the contribution chain to compensate for the creation of the new link.

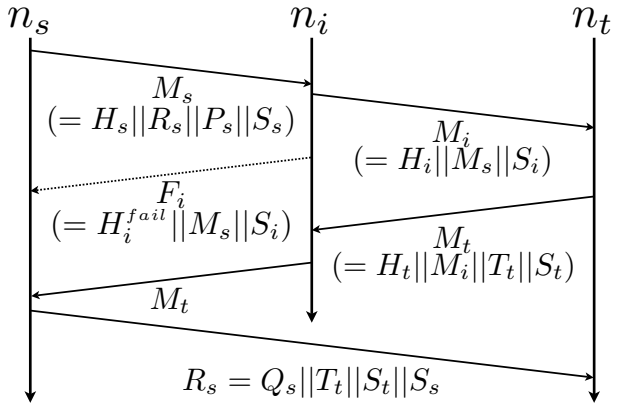


Figure 4.2: The contribution transfer operation

The basic operations that *PledgeRoute* uses to implement a contribution transfer event are shown on Figure 4.2, and are similar to those in [273] and [309]. As a first step, a peer n_s issues a *Contribution Transfer Request Message* (CTRM) M_s. These are structurally very similar to PAMs, including the unreciprocated peer contributions w_{ij} and the self-certifying identifiers of all peers that they traverse, as well as their nested digital signatures.

In greater detail, each CTRM includes a transaction header H_s detailing the transaction being requested and a timestamp, and a request R_s containing the maximum and minimum amount of contribution that n_s wishes to transfer to n_t, as well as the path that this will request take. As part of its self-certifying identifier, M_s will also include its public key P_s. The CTRM is then signed and sent.

Each peer on the transaction path (here represented by a single intermediary n_i) will check the nested signatures in the trust chain, and prepend a header H_i containing the maximum amount of con-

[3]Of course, $\tau \leq \min(w_{ij}, w_{jk})$ and the maximum τ feasible on the contribution network coincides with the maximum flow over the simple path along the unreciprocated contributions chain, which makes it sybilproof (see Section 4.7).

tribution that it is willing to devote to the transaction. Again, it will sign the concatenation of H_i and M_s before forwarding it on to the next peer in the path. If the minimum contribution transfer requested by n_s exceeds the maximum amount that a given intermediary is willing to reserve, a failure message F_i is propagated back through the trust chain, in order to free the reserved trust amounts in previous peers. As is common with soft-state reservation protocols, the reserved contributions on each one of the peers in the contribution chain will be automatically released if the end-to-end transaction is not successfully completed within a timeout interval t_{T}. In case of protocol failure, a failure message propagates back along the contribution chain, in order to free the reserved contribution amounts in previous peers without having to wait for t_{T} to elapse.

When the CTRM reaches its destination peer n_t, it determines the amount of trust that can be shifted through the path, inserts this information on a header H_t and concatenates it with the received request M_i. Then, it appends a trust granting ticket T_t for the request R_s - this will allow n_s to increase its account w_{st} on n_t, and it includes its self-certifying identifier. Before sending this new message to n_i, n_t signs $H_t||M_i$ and T_t (independently).

As M_t propagates back through the trust chain, every peer checks its own included signature, and the signature of the peer it got the message from. Finally, when n_s receives the ticket T_t, it appends it to a service request Q_s and signs both. When n_t receives this message, it checks its own signature and increases w_{st}.

Once contribution has been transferred, it is indistinguishable from direct contributions, and peers are able to decide, with complete flexibility, where to "spend" the accumulated pledged resources that other peers maintain for them.

4.4.3 Contribution Transfers as a Routing Problem

By using the routing algebra model of [146], we can analyse the contribution transfer operations formally. In particular, we see that the routing of CTRMs according to maximum transfer capacity paths can be modelled with a routing algebra where $\min$ is applied over the edges of paths to calculate their capacity, and $\max$ is used to compare the desirability of different paths, preferring paths with greater capacities. Formally, our system can be modelled as the *max-min semiring* over the real numbers, $(\mathbb{R}, \max, \min)$, which is normally used to model bandwidth-aware routing.

A useful property of this algebra is that it is *monotonic*: the addition of links to paths preserves the ordering of paths in terms of their desirability. This means that if we have two paths P_{ij} and P_{ik} starting from n_i, and P_{ij} is weakly preferred over P_{ik} (denoted as $P_{ij} \succsim P_{ik}$), then prepending a new path P_{si} to both P_{ij} and P_{ik} to form two alternative paths $P_{sj} = P_{si} \cup P_{ij}$ and $P_{sk} = P_{si} \cup P_{ik}$ will preserve the ordering (desirability) of their path transfer capacities, and we have that $P_{sj} \succsim P_{sk}$. This, in turn, allows n_i to find optimal contribution transfer paths incrementally, by advertising all incoming links and their weights w_{jk} to all its neighbours in $\mathcal{G}$, and running a Dijkstra-like algorithm [266] to route over the contribution network $\mathcal{G}$. In our case, this "link flooding" approach is avoided both for increased scalability and to preserve the slander resistance properties of the system (see Section 4.7), but each peer n_i uses a Dijkstra-based algorithm locally over its sampled topology $\mathcal{G}_i$ to determine transfer paths.

4.5 Trust and Contribution Cycle Seeding

It is clear that the contribution transfer process detailed in Section 4.4.2 drains the contribution network $\mathcal{G}$, as it subtracts τ from all links in the CTRM path, while only producing a single new one of weight τ between the transaction originator and the last peer in the chain. Although this is not harmful to peers, because they still have the same wealth (their contributions are reduced in the same amount as their pledges), it is deleterious to the capacity of the network to perform contribution transfers, as it depletes the link capacities that make it possible.

There are two processes that can be used to counter this effect, by injecting social capital into $\mathcal{G}$. The most fundamental one is *true altruism*: when peers give contributions without decreasing any account in $\mathcal{G}$, there is a net increase in social capital. The second process is *mutual crediting*, in which an hypothetical contribution of value zero is "decomposed" into two nonzero contributions between two peers, but with opposite directions. However, just as *Tit-for-Tat* can be generalised to cyclical multiparty exchanges, mutual crediting can be extended to any closed cycle of consistently oriented links in $\mathcal{G}$. None of the peers involved would experience an increase in wealth, but a number of links with nonzero weights would be added to $\mathcal{G}$.

To accomplish this, we present a process by which peers can find sets of other peers that can agree to create new contribution cycles between them. This means that each one of the peers in the set will pledge some resources to a *successor* peer and gain a pledge for the same amount from a *predecessor* peer, in a ring of shared obligations. Since the wealth of every peer in one of these contribution credit cycles remains unchanged, the only effect of this operation is to replenish link contributions and facilitate contribution transfer.

Naturally, each peer must be confident that the next peer in the cycle will actually reciprocate these virtual contributions in case it is requested to do so. Thus, every peer must estimate the amount of contributions that it can be confident that each of its neighbours would eventually pay back with high probability: the *safe credit margin* υ_{ij} defined on Section 4.2.

We calculate υ_{ij} as a monotonically increasing, sub-linearly growing function of $\hat{w}_{ij}$ (recall from Section 4.2 that the $\hat{w}_{ij}$ are the contributions that n_j has reciprocated to n_i). The reason for this is that, in order to make the safe credit margin reliable, peers will demand an increasing amount of successfully reciprocated contributions in order to increase υ_{ij}. We present one such function, where the increase in $\hat{w}_{ij}$ required to allow for an increase in υ_{ij} levels grows linearly. To present the intuition behind this function, let us consider two new peers n_i and n_j, such that $w_{ij} = w_{ji} = 0$ and $\hat{w}_{ij} = \hat{w}_{ji} = 0$. Since n_i is unable to ascertain the trustworthiness of n_j (and vice-versa), they will only provide each other with a basic amount of altruistic service: $\upsilon_{ij} = \upsilon_{ji} = \kappa$.

If peer n_i now gives altruistic contributions to peer n_j amounting to κ, then $w_{ij} = u_{ij} = \kappa$, and no more contributions are possible from n_i to n_j until reciprocation takes place. When n_j reciprocates by providing a service worth κ to n_i, n_j will set $w_{ij} = 0$, and n_i will set $\hat{w}_{ij} = \kappa$ and $u_{ij} = 2\kappa$. Increasing u_{ij} from 2κ to 3κ, however, will require a further increase of $\hat{w}_{ij}$ of 2κ. In general, an increase of u_{ij} from $n\kappa$ to $(n+1)\kappa$ will require an increase of $n\kappa$ in $\hat{w}_{ij}$. Thus, $u_{ij} = n\kappa$ will require $\hat{w}_{ij} = \kappa + 2\kappa + 3\kappa + \cdots + (n-1)\kappa$, and we have that the safe credit margin u_{ij} can be calculated as

$$u_{ij} = \min\left(\tfrac{1}{2}\left(\kappa + \left(\kappa^2 + 8\kappa\hat{w}_{ij}\right)^{\frac{1}{2}}\right), \xi_i\right),$$

where the first term inside the $\min$ is the positive root of $\hat{w}_{ij} = \frac{(u_{ij}-\kappa)u_{ij}}{2\kappa}$ (by the summation above), and ξ_i is the maximum credit allowed by n_i, as defined in Section 4.2.

Each peer then advertises their υ_{ij} values to their neighbours in a manner analogous to the w_{ij} values in PAMs. The messages used for this purpose are called *Credit Announcement Messages* (CAMs), and they also include nested digital signatures of the peers that have received and forwarded them. This time, however, the random walk probabilities of the CAM are weighted by the υ_{ij} of the neighbours, instead of their w_{ij}. Each peer n_i that receives a CAM will use it to update its model of the credit topology with an oriented path P_{si} of safe credit margins starting on the message originator n_s and ending on n_i itself.

As the information present in many CAMs is accumulated in n_i, it will create a local model $\mathcal{G}\{\upsilon\}_i$ of the safe credit network $\mathcal{G}\{\upsilon\}$. Of course, since all paths obtained from CAMs terminate in n_i, $\mathit{\Pi}^-(n_i) =$

$\emptyset$. Another consequence of this is that we can be sure that for every peer $n_j \in \mathcal{G}\{\upsilon\}_i$, there is at least one directed path P_{ji}. Thus, if n_i can estimate its safe credit margin υ_{ij} to n_j and through this create a new link l_{ij} with weight υ_{ij}, it can create a cycle c_{ij} by concatenating P_{ji} with l_{ij} (obviously, $l_{ij} \circlearrowleft c_{ij}$ and $l_{kl} \circlearrowleft c_{ij}$ for all $l_{kl} \in P_{ji}$). Of course, the maximum amount of credit that can be consistently applied over the entire cycle is equal to the minimum credit that can be accepted on any one of its constituent links, so that the maximum credit $\tau_c = \min(\upsilon_{ij}, \{\upsilon_{kl}\} : (n_k, n_l) \in P_{ji})$. Informally, if the peer can estimate its safe credit margin to any of the peers present in its credit topology $\mathcal{G}\{\upsilon\}$, the new link that this assessment creates will generate a correctly oriented cycle on the credit graph, and the minimum credit on its constituent links will determine its value. Then, using a procedure analogous to the one used for contribution transfer, the peers can perform the necessary additions to their accounts in order to seed the contribution network $\mathcal{G}$.

4.6 Incentive Mechanism

4.6.1 Account Maintenance Incentives

Since most of the protocol elements that we propose involve the consistent modification of the account values w_{ij}, protocol stability in the presence of strategic peers demands that peers have no incentive to arbitrarily change the accounts relating to the contributions of other peers.

As detailed in Section 4.4.1, every time that a peer n_i sends a PAM (either newly created or forwarded) to another peer n_j, n_j can check if the value that n_i reports of w_{ji} corresponds to the actual contributions c_{ji} from n_j to n_i. The most straightforward way of dealing with this scenario would be for n_j to reduce w_{ij} by $c_{ji} - w_{ji}$ if $c_{ji} > w_{ji}$, and to increase w_{ij} by $w_{ji} - c_{ji}$ if $w_{ji} > c_{ji}$ (as long as this does not bring $w_{ij} - w_{ji}$ over the safe credit margin υ_{ij}). This policy directly mimics the reciprocity properties of the *Tit-for-Tat* trigger strategy in an iterated *Prisoner's Dilemma*, and as such it can take advantage of the significant literature available [190] on its convergence, evolutionary stability, and how to deal with its vulnerability to observation noise (misjudged defections triggering mutual defection runs).

4.6.2 Protocol Incentives

The protocols described in Section 4.4 presuppose the correct operation of the contribution accounting system, which is itself vulnerable to freeloading. To see why, it suffices to consider the benefit that a strategic peer might obtain from message forwarding. In the case of a PAM, a peer n_i does not need to forward the PAM to obtain a flow contribution path rooted on itself. Thus, by receiving PAMs but refusing to forward them, a peer can conserve its resources and still benefit from contribution topology discovery.

To provide a negative incentive to the termination of PAM walks, every peer decreases the account of the next peer in the contribution chain by an amount χ_{PAM} before sending the PAM. Consider a PAM traversing the path in Figure 4.1a. In this case, w_{ji} would be decremented by χ_{PAM}, giving n_i a profit of χ_{PAM} and n_j a deficit for the same amount. However, when n_j forwards the PAM to n_k and w_{kj} is decremented, n_j will regain its original level of contribution and it will be n_k who has a deficit. In this way, the deficit of χ_{PAM} is propagated through the contribution chain, until the PAM is returned to n_i (and its initial profit cancelled). This means that, if the PAM fails to be forwarded (or returned to the originator), the last peer that received it will automatically suffer a contribution fine of χ_{PAM}.

For CTRMs, any peers not at the contribution chain endpoints have nothing to gain from the transaction, but still have to consume their own resources. Again, any peer could choose not to propagate CTRMs from different peers, and it would save resources without impacting its own transfer capacity.

To create an incentive for contribution transfer transactions to be correctly executed, peers rely on

the transaction timer t_{T}. The expiration of t_{T} triggers two events. First, the soft-state reservation for the transaction is freed. Second, each peer automatically decrements the account of the next peer in the contribution chain by an amount χ_{CTRM}. The account movements balance in the same way as in the PAM case, but instead of equalising the contribution contributions, the peer that failed to complete the transaction is fined with χ_{CTRM}, and the transaction originator is given a compensation of χ_{CTRM}.

4.7 Attack Resistance Properties

4.7.1 Resistance to Sybil Attacks

First, we show that our proposed system is resistant to *sybil attacks* [106]. In our case, the sybil attack takes the following form: a peer n_m creates a large set of identities, and directly modifies their account values to create an arbitrary contribution network among them (a *sybil strategy*, as defined in [71]). Then, it tries to use this fictitious contribution network to extract large amounts of resources from the network.

We argue that this attack will not be profitable for the attacker. The reason for this is the equivalence between contribution transfer and maximum flow over simple paths in $\mathcal{G}$. This implies that the capacity of this "network of sybils" to extract resources from the peer-to-peer infrastructure remains bounded by the contribution values of the links that connect it to the rest of the system (the set of *attack links*, as defined in [323]). As these will be, in turn, bounded by the contributions that the peers in the sybil network make to normal peers, there is no incentive to create large numbers of identities: this will just split the contribution values amongst all of them, and no single one of them will benefit. Every one of the sybils will experience worse service than a single identity with access to the same network resources.

To analyse the sybilproofness properties of contribution transfer, we use the theoretical framework presented in [71] and in Section 3.2. We define the maximum contribution transfer capacity between a peer n_s and another peer n_i as

$$\mathcal{W}(s,i) = \bigoplus_{P_{si} \in \mathscr{P}_{si}} \left(\bigotimes_{l_{jk}=(n_j,n_k) \in P_{si}} w_{jk} \right), \tag{4.3}$$

where $\mathscr{P}_{si} = \{P_{si}\}$ is the set of all paths from n_s to n_i, and each of the l_{jk} is a contribution link in $\mathcal{G}$. Since in our particular routing algebra $\otimes = \min$ and $\oplus = \max$ (see Section 4.4.3), we define $\mathcal{W}(P_{ij})$ simply by applying $\min$ to the w_{nm} values of the $l_{nm} \in P_{ij}$.

It is clear that (4.3) is a special case of the anchor-flow reputation function (3.1), where *each peer anchors its "reputation" flow on itself*. Thus, rather than having a global anchor node that every peer uses to measure "reputation" from, each peer can rank each other peer in terms of how much contributions can be transferred to it through $\mathcal{G}$. More formally, we have that

$$g(P_{si}) = \bigotimes_{l_{jk} \in P_{si}} w_{jk} = \min(\{w_{jk} : l_{jk} \in P_{si}\}),$$

where $\oplus = \max$ is used in practically the same way as in (3.1), and the $\max$ operation over the power set $\mathbb{P}_{ai}$ is superfluous, since by monotonicity (see below) the set $\mathscr{P}_{si}$ of paths that maximises $\mathcal{W}(s,i)$ is known - it includes every possible path between n_s and n_i.

From these definitions, it is easy to see that our system satisfies the conditions for both value and rank sybilproofness (Theorems 4 and 5 of [71], and Section 3.2.3):

1. *Diminishing returns*. If we have a path P_{ij} with capacity $\mathcal{W}(P_{ij})$ and we concatenate it with another path P_{jk} with transfer capacity $\mathcal{W}(P_{jk})$ to create a new path P_{ik}, we have that $\mathcal{W}(P_{ik}) = \mathcal{W}(P_{ij} \cup P_{jk}) = \mathcal{W}(P_{ij}) \otimes \mathcal{W}(P_{jk}) = \min(\mathcal{W}(P_{ij}), \mathcal{W}(P_{jk})) \leq \mathcal{W}(P_{ij})$. It follows that

$\otimes = \min$ is nonincreasing.

2. *Monotonicity*. If we have two paths P_{ij}^1 and P_{ij}^2 with capacities $\mathcal{W}(P_{ij}^1)$ and $\mathcal{W}(P_{ij}^2)$, and we aggregate them with $\oplus = \max$, we have that $\mathcal{W}(P_{ij}^1) \oplus \mathcal{W}(P_{ij}^2) = \max(\mathcal{W}(P_{ij}^1), \mathcal{W}(P_{ij}^2)) \geq \mathcal{W}(P_{ij}^1)$. Then, we have that $\oplus = \max$ is nondecreasing.

3. *No splitting*. We consider a path $p_{in} = \{l_{ij}, l_{jk}, \ldots\}$ that has a contribution transfer capacity of $\mathcal{W}(P_{in}) = \bigotimes_{l=(n_i,n_j)\in P_{in}} w_{ij} = \min(w_{ij}, w_{jk}, \ldots)$. We split this path into two parallel paths P_{in}^1 and P_{in}^2 that span the same links as P_{in}, but which have link capacities w_{ij}^1 and w_{ij}^2 such that, for every link l_{ij}, $w_{ij}^1 + w_{ij}^2 = w_{ij}$. Since we require $w_{ij}^1 \geq 0$ and $w_{ij}^2 \geq 0$, we have that $w_{ij} \geq w_{ij}^1$ and $w_{ij} \geq w_{ij}^2$ for all l_{ij}. Then, we have that $\mathcal{W}(P_{in}^1) \leq \mathcal{W}(P_{in})$ and $\mathcal{W}(P_{in}^2) \leq \mathcal{W}(P_{in})$, and it follows that $\mathcal{W}(P_{in}^1) \oplus \mathcal{W}(P_{in}^2) = \max(\mathcal{W}(P_{in}^1), \mathcal{W}(P_{in}^2)) \leq \mathcal{W}(P_{in})$.

Furthermore, since contribution is transferred through single routes and $\oplus = \max$, contribution transfer is rank sybilproof.

4.7.2 Resistance to Peer Slander

Our system is also designed to mitigate peer *slander*. One case of this problem involves peers lying about the contributions that they have received from other peers. As noted in Sections 4.4.1 and 4.4.2, the effectiveness of this attack is reduced because all information concerning the contribution of a peer must be vetted by its neighbours before being passed on.

Another, related case involves peers lying about the contributions that they have given to other peers. Again, this is of limited impact in our system, since it would entail forging a digital signature.

4.7.3 Resistance to Whitewashing

Finally, we show that our system is resistant to *whitewashing* attacks: the creation of disposable identities, so that any penalties that the system might have imposed on misbehaving peers are "forgotten". Since the only way to make whitewashing unprofitable is to make a newcomer and a heavily punished node indistinguishable, this attack is difficult to defend against: it exploits altruistic system characteristics that are desirable for bootstrapping and incorporating new peers into the system.

In our system, all punishments are equivalent to the loss of previous contributions. In effect, both newcomers and heavily punished peers have zero social capital. Since these peers are unable to engage in reciprocative interactions, they will only receive best effort service (they will be easily preempted by peers with higher contribution values) and will not be able to participate in the contribution transfer network. To see why, it suffices to recall that the PAM random walk is weighted by previous contributions: a node with very small previous contributions has a very small probability of being forwarded a PAM. As peers increase their contributions, they become more heavily embedded in the contribution network and their capacity for contribution transfer increases. However, we note that while non-malicious peers will only need to build up their contributions once, whitewashers will need to do so several times: once for each one of their new identities. This provides peers with an incentive to maintain their identities and behave correctly towards other peers.

4.8 Simulation Experiments

We analyse our system by performing extensive simulations. When we use a synthetic contribution topology $\mathcal{G}$ to test our protocols, we use Erdős-Rényi [53] graphs, and if weighting is required, we assign link weights following a uniform distribution. We simulate each peer individually, in order to take into account differences between the true contribution network $\mathcal{G}$ and its sampled model on each peer, $\mathcal{G}_i$.

4.8.1 Contribution Topology Discovery

In order to get a better understanding of the behaviour of our accounting mechanism, we track the formation of the contribution network subgraph $\mathcal{G}_i$ for each of the peers as the PAMs propagate through the contribution network. To evaluate this, we use a static $\mathcal{G}$ with 1000 peers, each with an average of 150 neighbours. We leave the analysis of the influence of dynamic changes to these parameters for further work.

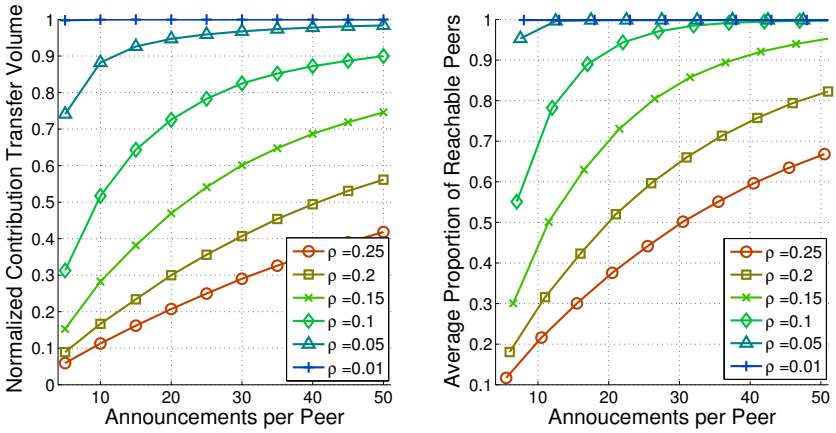


Figure 4.3: Contribution transfer volume and reachability in $\mathcal{G}_i$

As expected from our proposed PAM forwarding probability (4.1), the locally constructed subgraphs $\mathcal{G}_i$ tend to be much sparser than $\mathcal{G}$ and may fail to account for all possible paths through which contribution transfers could be routed. To analyse this, we consider the *normalised contribution transfer capacity*: the sum of all the potential contribution transfers that n_i could perform over $\mathcal{G}_i$ using the routing algebra of Section 4.4.3, normalised by the same magnitude calculated over the entire contribution network $\mathcal{G}$. As we can see in Figure 4.3 (left), not many messages are needed to attain a high capability for contribution transfer. With just 10 PAMs per peer and a ρ_{stop} of $.1$, the average contribution transfer capacity of a peer is nearly at 55% of its theoretical maximum.

If we focus not on the capacity of the contribution transfers, but on the existence of a contribution transitivity path on $\mathcal{G}_i$, our results are even better. As shown in Figure 4.3 (right), the proportion of peers reachable through contribution transfers grows quickly with the number of PAMs per peer (its growth rate decreases, as expected, as the walk length approaches the graph diameter). The standard deviation of the distribution of both the normalised contribution transfer capacity and the number of peers reachable through the discovered contribution transfer paths decreases with increasing PAMs per peer, as $\mathcal{G}_i$ approximates $\mathcal{G}$. The range in the number of announcements per peer in Figure 4.3 was selected to be small enough to be practical in conventional Internet deployments of *PledgeRoute*.

It is interesting to consider the cumulative distribution function (CDF) of the capacities of all the contribution transfers that peers can achieve using their locally constructed $\mathcal{G}_i$. We compare it with the CDF of all the contribution transfer capacities that could be achieved if every peer had perfect knowledge of $\mathcal{G}$, represented as a dashed line in Figure 4.4 (due to the construction of our simulated $\mathcal{G}$, most peers tend to achieve roughly the same contribution transfer capacity between them). As the number of PAMs increases, $\mathcal{G}_i$ samples $\mathcal{G}$ more accurately and the CDF of locally discovered contribution transfer capacities becomes much closer to the theoretical maximum.

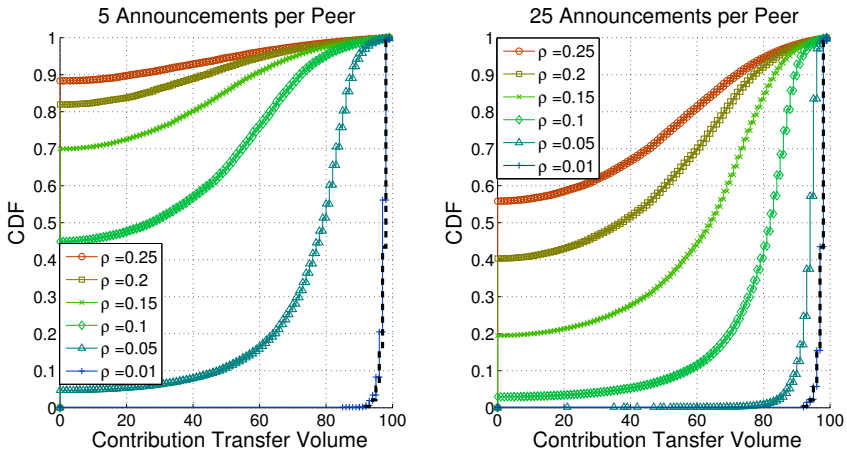


Figure 4.4: Change in transfer capacity CDF with number of PAMs

4.8.2 Contribution Transfer and Indirect Reciprocity

To evaluate the properties of our system as an incentive mechanism,we use an *interest digraph* $\mathcal{H}$ to model peer service preferences: an edge (n_i, n_j) in $\mathcal{H}$ implies that peer n_i is interested in requesting services from peer n_j.

We define two types of peers: *reciprocators*, that follow the reciprocity rules described in Sections 4.4, and *freeloaders* that rely exclusively on altruism and refuse to honour any service requests. Moreover, we distinguish two kinds of reciprocators: *enabled* peers are able to transfer contributions and perform contribution seeding as described in Sections 4.4 and 4.5[4], while *non-enabled* peers can only perform direct reciprocity (strict *Tit-for-Tat* with peers with whom they have interacted in the past).

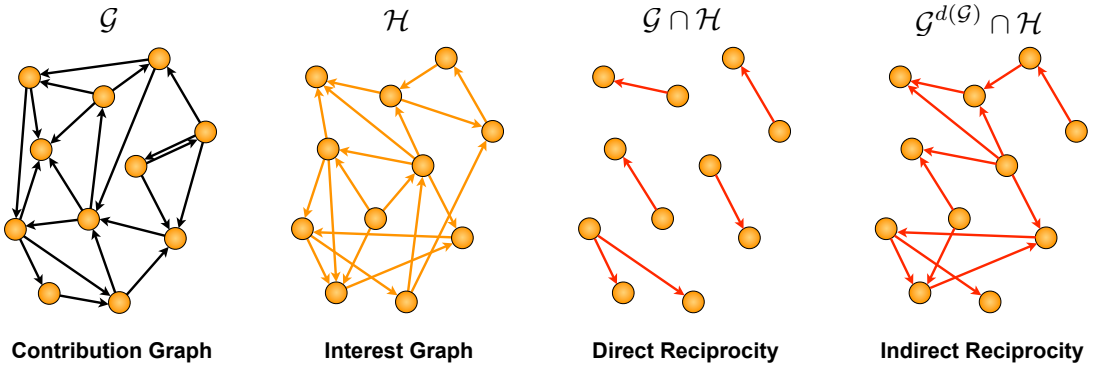


Figure 4.5: Indirect reciprocity and the service capacity improvement it provides.

The significant improvement that indirect reciprocity provides over direct reciprocity is evident from Figure 4.5, where $\mathcal{G}$ and $\mathcal{H}$ are examples of contribution and interest graphs. The links in $\mathcal{G} \cap \mathcal{H}$ are those services that could be achieved using direct reciprocity: those that imply the existence of both interest and accumulated social capital between the two peers involved. If we recall $\mathcal{G}^{d(\mathcal{G})}$ (Definition 3.18), we see that the links in $\mathcal{G}^{d(\mathcal{G})} \cap \mathcal{H}$ are those service interest links in $\mathcal{H}$ that can be achieved by routing contributions over $\mathcal{G}$. Since obviously $\mathcal{G} \subset \mathcal{G}^{d(\mathcal{G})}$, indirect reciprocity allows an improvement of the service delivery capacity of a peer-to-peer system. Of course, the reachability that routing over $\mathcal{G}$ provides will be a function of its structure and how well matches that of $\mathcal{H}$. If one assumes that $\mathcal{G}$

[4]When an *enabled* peer requests a service from a peer to whom it has not contributed, it will attempt to execute a contribution transfer operation. When contacted by a peer to whom they owe past contributions, *reciprocators* will provide prioritised service (all other requests will only get best-effort service). When a freeloader receives a request, it will ignore it.

and $\mathcal{H}$ are engineered to be similar (as in [255]), it may be that a small number of hops in $\mathcal{G}$ are enough for good coverage in $\mathcal{H}$, which in turn could be useful in decreasing routing table size. We leave such optimisations for further work.

In order to focus exclusively on the reciprocity properties of the system, we neglect peer capacity: all peers will respond to all requests either with high quality service, best effort service, or refusal due to freeloading behaviour. We are interested in exploring how the service quality experienced by reciprocators and freeloaders varies as $N_{\mathcal{H}}$, the average peer outdegree in $\mathcal{H}$, and $\frac{R}{F}$, the proportion of reciprocators vs. freeloaders, change.

Algorithm 1 Execution of a *simulation epoch* for a single time t

for all $n_i \in \mathcal{G}$ (in random order) **do**
  for all service requests s_i of n_i (10 in our simulation) **do**
    Calculate S_i, the set of peers in $\mathcal{G}$ known to n_i that are of interest in $\mathcal{H}$ ($S_i \subset \mathcal{G}_i$, $S_i \in \Pi^{-}(n_i)$ in $\mathcal{H}$)
    Rank all $n_s \in S_i$ using s_1, then tie-break ranking with s_2, then tie-break with *random*
    Send request s_i to the peer n_s that had the best ranking
  end for
end for

We start each *simulation run* with peers having no contributions between them, and then allow the simulation to run until an equilibrium is reached where the average proportion of services that a peer receives through reciprocation and altruism remains unchanged. On each one of the epochs of a simulation run, the actions in Algorithm 1 are performed, where s_1, s_2 and *random* represent peer selection strategies for each peer n_i:

- s_1: Prefer peers to whom n_i has contributed, and are thus "indebted" towards it.
- s_2: Prefer peers that have reciprocated towards n_i in the past.
- *random*: Choose any peer randomly.

Finally, we run a simulation run for each possible combination of $N_{\mathcal{H}}$ and $\frac{R}{F}$, and terminate the simulation only after the average service quality for all peers in $\mathcal{G}$ has converged. The results are shown in Figure 4.6.

In Figure 4.6, blue circles represent *reciprocators* and red triangles represent *freeloaders*. The left column shows the results obtained when *reciprocators* are *non-enabled*, while the right column shows those obtained when *reciprocators* are *enabled*. Finally, the top row shows the proportion of time that peers obtain high priority service, while the bottom row shows the proportion of time that peers obtain best effort service[5]. We plot the average service quality that a peer receives, as a function of $N_{\mathcal{H}}$ normalised by the number of peers in $\mathcal{H}$. The error bars mark the maximum and minimum values obtained by varying $\frac{R}{F}$ between .5 and .9, for each value of $N_{\mathcal{H}}$.

We can see that both *enabled* and *non-enabled* reciprocators obtain a higher ratio of prioritised service as $N_{\mathcal{H}}$ increases, since this gives them a better chance of being able to find a peer to whom they can send a reciprocating request. However, *enabled* peers experience a much higher proportion of prioritised service requests, and this proportion increases much more quickly than for *non-enabled* peers. This is because contribution transfer allows *enabled* peers to perform reciprocative interactions almost exclusively, being able to route their contributions in $\mathcal{G}$ to match the supply and demand constraints imposed by $\mathcal{H}$. Unable to do this, *non-enabled* peers are forced to maintain higher proportions of their service requests as altruistic ones, thus suffering a service quality penalty. Additionally, we see that freeloaders

[5] Due to freeloaders, some requests will not be honoured and these two rows will fail to add up to one.

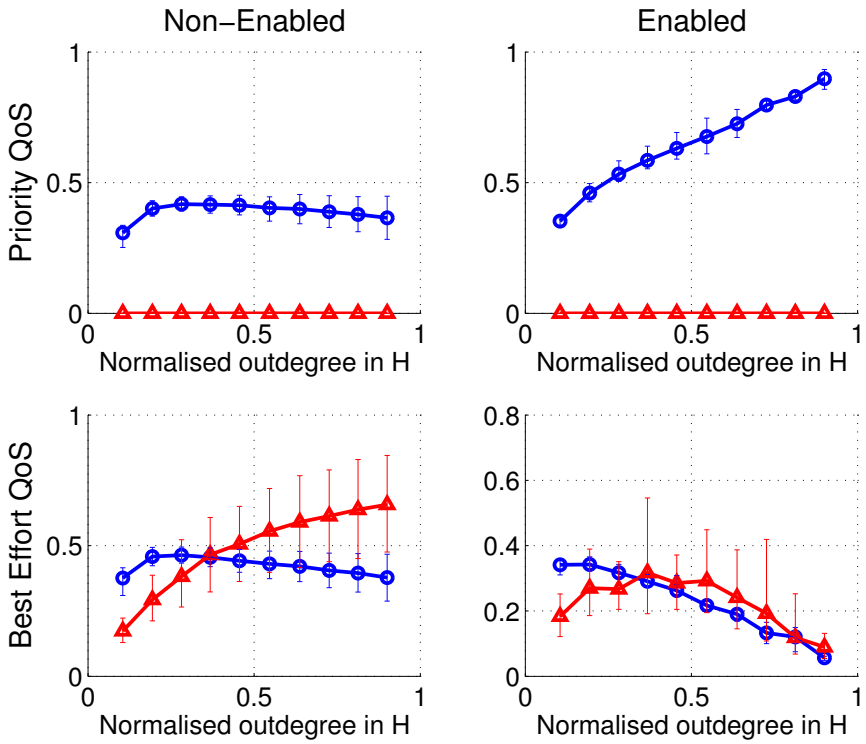


Figure 4.6: Average service quality as a function of $\frac{N_{\mathcal{H}}}{|\mathcal{H}|}$

experience consistently worse service levels and much higher variability, validating the effectiveness of reciprocity as an incentive mechanism.

4.9 Using *PledgeRoute* as a Payment Mechanism

Although the contribution transfer operation has been presented as a sybilproof protocol operation enabling indirect reciprocity, it has wider applicability in the context of the design of incentive mechanisms for peer-to-peer networks. In particular, it can be used as a completely decentralised *currency platform*, where peers use self-minted local currencies that are exchanged through the contribution transfer operation (called *currency transfer* in this case). Instead of being backed by a central bank, this currency is *individually backed by each peer*, with the network $\mathcal{G}\{\hat{w}\}$ of reciprocated contributions $\hat{w_{jk}}$ operating like a *web of trust* that can be used to set up bootstrapping credits through cycle seeding and direct altruistic requests.

Two basic operations are defined for the currency platform along with currency transfer: currency *deposit* and *withdrawal*. A deposit is mapped to granting an altruistic service request, and in effect it increases the wealth of the peer granting the service, while the peer to whom the service is being granted increases its obligations (for this peer, we call this operation a *buy*). On the other hand, a withdrawal is mapped to being granted a reciprocative request, and it decreases the wealth of the peer requesting the service. The peer granting the reciprocative request decreases its obligations, and for this peer we call this operation a *sell*. This is shown graphically in Figure 4.7 (for an example of how could this be used in an auctions-based system, see Section 5.4).

Although the mapping between social capital and currency seems transparent, there is a fundamental difference: is there is no longer a need for a fixed exchange rate between contributions and currency units. Thus, the use of a currency system enables peers to use a pricing mechanism to assign value to candidate service options. This property of a currency system will be of great use in Chapters 5 and 6, where the

Figure 4.7: *PledgeRoute* as a payment system

valuation that peers give to services becomes of critical importance[6].

One possible weakness of *PledgeRoute* when used as a currency platform is that each transaction will be reflected in two different accounts, and there is a risk that these will not match if peers do not agree in the correct value of a service. In this case, the oversight mechanisms detailed in Section 4.6.1 can be used to manage and resolve such discrepancies.

Although currency transfer is not enough, by itself, to implement an incentive mechanism, it is a very useful component for more complex incentives systems. When supplemented with statistical throughput models, it can be used to guard against hidden action in QoS overlays (see Chapter 6), and when supplemented by an auction-driven resource allocation mechanism it can be used to streamline peer-to-peer swarming and content distribution (see Chapter 5).

4.10 Discussion of Potential Improvements

In this model we presented *PledgeRoute*, a decentralised accounting system that, when used in conjunction with reciprocity policies and two classes of service (*altruistic* and *reciprocative*), operates as an incentive mechanism.

Since *PledgeRoute* is based on reciprocity, it will be useful in more or less symmetric systems, where contributions are sometimes given and sometimes received by all the peers. Thus, an obvious improvement path would be to consider a *PledgeRoute* alternative for heavily asymmetric systems, such as conventional Internet TV broadcasting, where instead of a peer-to-peer network where all peers are equivalent and function as servers and clients in approximately the same ratio, we have a tree-like distribution structure originating from a single, infrastructure location which does not download at any point. Under such circumstances, reciprocity seems a poor fit, and different incentive models are required.

[6] Since pricing and valuation systems can be complex and varied in their implementation details (see for instance [161, 73, 119] and Chapter 5), we do not require any particular one for *PledgeRoute* when used as a payment system.

5

An Incentive Mechanism for Service Differentiation using Vickrey Auctions

One interesting aspect of using a completely decentralised, self-organising peer-to-peer overlay for real-time streaming is that it provides essentially unlimited scalability without the need for infrastructure support. This of course can entail significant communication and coordination costs and pose difficult privacy issues, but its scalability and resistance to single points of failure can make it attractive even in light of these challenges (see Section 2.3).

We define a *swarming protocol* as a set of rules that allow each peer to determine, in a completely decentralised fashion, to which other peers will it send service requests, which service requests from other peers will it accept, and the service quality that it will give to each one of these service requests when they are served.

Protocols of this kind implement tradeoffs between system quality metrics while respecting peer capacity constraints. This, however, is often complicated by factors such as service availability under churn, complex content search issues, varying achievable QoS between different peers and incentives issues. In our case, we restrict our attention to the design of a fully distributed resource allocation protocol that will give increasingly better QoS as peers contribute an increasing amount of resources to the overlay.

Due to the great complexity of the aforementioned tradeoffs, swarming protocols tend to either make simplifying assumptions or address only subsets of the problem. We now explain where our proposed protocol stands in these issues.

5.1 System Overview

We commence by recalling Property 1 in Chapter 1 (The *Fundamental Property of Incentive Mechanisms*), and thus defining an *incentive mechanism* as a resource allocation policy where the utility that each participating strategic peer n_i obtains is directly proportional to the utility that the other participating peers achieve because of the participation of n_i in the overlay. At its most basic, it is a protocol in

which peers allocate their resources preferentially to those peers that provide more of their own resources for use in the peer-to-peer overlay. Of course, this means that different peers must have a way to affect the utility that other peers get from the system, depending on the amount of resources they provide to other peers.

To achieve this, we require that peers have a *quality differentiation* policy in place. This means that they can allocate resources to peers in such a way that any particular ordering of peers in terms of decreasing quality can be achieved with minimal wasted or unused resources. This will be heavily dependent on QoS measurement and enforcement particulars, but obvious examples of tools to achieve this include packet prioritisation, differentiated queueing, query service order sorting or differentiated capacity assignment.

5.1.1 Assumptions

We commence by laying out the groundwork for the analytical contributions that we present in this chapter by presenting our basic assumptions. As is usually the case, these will be motivated not only by our desire to achieve an accurate model, but also by the necessity for mathematical tractability.

We assume that time is slotted in intervals T, with each interval being called a *round*. We will use the auctions nomenclature presented in Section 3.4.2. We consider the multiple-item Vickrey auction when applied to a simplified peer-to-peer chunk swarming scenario. Each peer will behave, each round, as a server (an auctioneer) and a client (a bidder). Peers will only open one auction per round as auctioneers, but they will participate in all the auctions of their interest simultaneously.

Each peer n_i, in its role as an auctioneer, will have M_i *service units* in auction, with this number being contingent on its resource capacity. In addition, we assume that resource allocation for round k is decided during round $k-1$: during round $k-1$ all auctioneers receive bids from all interested peers, and all auctions are resolved at the end of that same round. Service unit delivery, as well as the next bidding round, commences at the beginning of round k.

We assume that these service units are *content independent*, so that their quality is only a function of the peers involved and not which specific content subset is being requested. This means that the value of each service unit will be a function of the properties of the peers involved and the overlay link between them. This is not due to an inherent limitation on Vickrey auctions - rather, it is a simplifying assumption that helps in obtaining closed-form solutions for some magnitudes of interest.

We assume that peer quality information will be collected by each peer, distilled into statistics and made available to other peers through a market system $\mathcal{M}$, which may also include accounting and payment components. In our case, we use *PledgeRoute* (see Sections 4.9 and 5.4) as an accounting and payment mechanism.

The various components of the incentive mechanism are presented in Figure 5.1. The three basic components are the *bidder*, *auctioneer* and *accounting* processes. The bidder process takes as input information from network measurements (network layer latencies, end-to-end throughput, synthetic coordinate information, etc.) and from the market system (bid value probability densities, average peer load, peer resource capacity, etc.) in order to determine from which peers to request service. These requests take the form of *bids* for resource auctions at other peers. When one or more of the bids of a peer are selected for service (we say that these bids *win*), the accounting process records this, updates relevant accounts accordingly, and informs the auctioneer process of the account balances for the peers whose bids win auctions, so that they can be assigned to a proper QoS class. The auctioneer receives bids from remote bidders, compiles bid probability densities and updates them in the market system along with the total number of received bids and the total amount of resources that the peer makes available to the overlay. Finally, the auctioneer process selects which bids will receive service, determines the quality class for each won bid, and provides services to the remote peers accordingly.

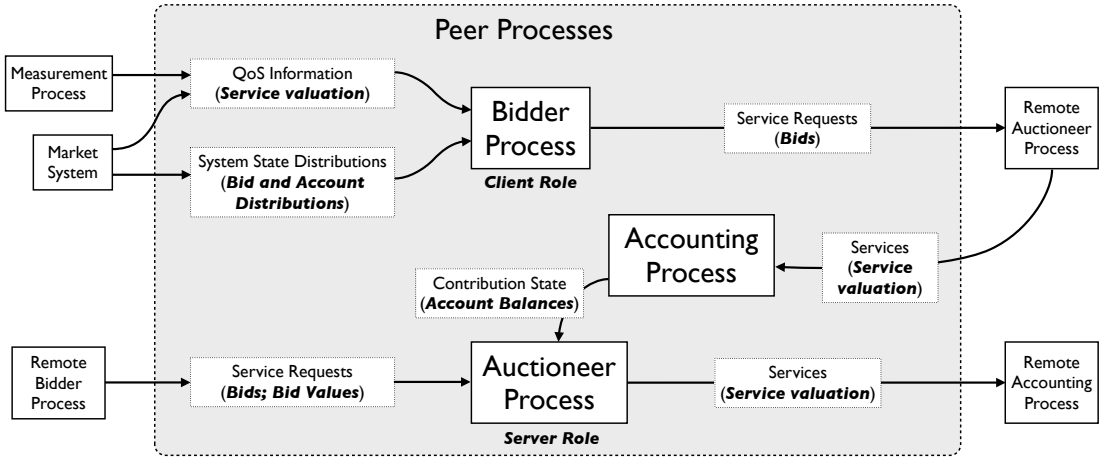


Figure 5.1: Auction-based swarming: system diagram

5.1.2 Peer Value Estimation

We assume that each peer p_j will have a mechanism to obtain the value v_{ij} of receiving a service unit from a given auctioneer n_i. Relevant variables that might affect service unit value include end-to-end throughput between n_i and p_j, network layer jitter, latency and loss between n_i and p_j, the total amount of CPU or upload bandwidth that n_i provides to the overlay, the transaction response time at n_i, or any other measure that might have an influence on the usefulness or desirability for p_j of a service unit from n_i. We give an example of how might this be accomplished for a specific incentives model in Section 5.5.1.

We do not prescribe any particular value estimation function as part of this model, allowing each peer to supply its own instead. This allows peers to freely express their preferences for services and make appropriate quality tradeoffs between candidate auctioneers.

5.2 The Auctioneer Process

For our analysis, we assume that within each peer n_i there is an auctioneer process auctioning M_i service units to the rest of the peers, which we denote as $\mathsf{P} = \{\mathsf{p}_j\}$. We will use a multi-item Vickrey auction, and thus the outcome k^* is determined in exactly the same way as in Section 3.4.2.3: each client peer p_j is assumed to have sent m_{ji} bids, each at a value $v_j(\zeta_j)$ where ζ_j corresponds to the *type* of p_j (we describe ζ_j in more depth in Section 5.3.1). The bids are ranked in increasing order of value, and the top M_i win a service unit. The calculation of the *Clarke pivot value* is trivial as well - it is just the sum of the winning bid values *had* p_j *not participated*. Therefore, the payment h_j that will be applied to the winning peers p_j is just the sum of the top X^{ij} losing bids, where X^{ij} is the number of bids that p_j won in a given auction by n_i. The utility U_j that p_j has (after payments) is just the difference between the sum of the values of its X^{ij} winning bids and the sum of the top X^{ij} losing bids.

We will make two simplifying assumptions. First, we follow the *independent private values* assumption in the sale of M_i service units to N_i bidders. Thus, the value that each bidder gives to the service units being auctioned is private, and uncorrelated with the values of the other bidders. Second, we will assume that peers experience *no synergy* in acquiring increasing numbers of service units. This means that for each p_j, the utility in obtaining X^{ij} service units from n_i is just X^{ij} times the utility in obtaining a single service unit from n_i. Therefore, there are no complementarity or substitution effects between different service units, and all service units from a given auctioneer have the same value. Formally, this means that the model only requires a single per-item valuation $v_j(\zeta_j)$ instead of a valuation

function $v_j(\zeta_j, X^{ij})$. In this case the player type ζ_j is a vector that includes quality-related properties of the peers and the overlay path between them and the auctioneer.

Throughout this chapter, we will be interested in estimations that peers make regarding the number of competing bids that they will have to face when they bid at other peers. Instead of taking a peer-independent point of view, we shall always consider bid load as relevant to the bid competition of a particular peer (which we usually denote as p_j). Thus, rather than the total number N_i of bids on n_i, we shall be interested in the number of bids in n_i with which the bids of p_j must compete - which usually involves removing from N_i those bids sent by p_j itself. We shall denote this number of bids as N_i^j, the *total competing load* of p_j at n_i.

We now define two basic load-related states for the auctioneer. We call the situation in which $M_i > N_i^j$ as an *overabundance* state, and $M_i < N_i^j$ as a *scarcity* state. In the scarcity case, even before p_j has sent a bid, there are more bidders than service units are in auction. In the overabundance state p_j can expect to have some of its bids accepted at zero cost. The reason for this is that following directly from the Clarke pivot rule, nonzero prices will only be charged in the scarcity state[1].

5.2.1 Calculating Expected Auction Outcomes

As detailed in Section 5.1.1, the main task for the auctioneer process will be selecting which received requests to select for service, and the main task for the bidder process will be to select which peers to select requests for which service units. Since it will be advantageous for the bidder to have a model of a potential auctioneer, we shall be interested in calculating the expected auction outcomes that a bidder will experience from a given auctioneer. We now turn our attention to this problem.

5.2.1.1 Estimating the Expectation of the Number of Successful Bids

It is interesting for a bidder p_j to know, given its valuation v_{ij}, the expected number of bids it will win with a particular auctioneer n_i. Since by the *incentive compatibility* property of the VCG mechanism we know that peers have no incentive to bid strategically, we assume that peers bid their true valuations. Thus, every peer p_j bids its true valuation v_{ij} for m_{ji} service units out of the total M_i being auctioned by n_i.

We simplify our protocol design by modelling the bid values as a random variable V^i with a probability density f^{V^i}, and sharing only the densities between the peers, rather than the actual bid logs. Then, $\int_{v_a}^{v_b} f^{V^i}(v)dv$ gives the proportion of valuations between v_a and v_b for all those bids that were submitted to n_i. Since players do not bid strategically, auctioneers can compile the true value probability distributions for their set of bidders from the bids that peers submit to them.

Again focusing on the prospective competition for peer p_j, in the model presented here we treat the N_i^j competing bids that n_i receives in a given interval T as a set of N_i^j realisations of the random variable V^i, which represents the valuations that all other peers have for service units provided by n_i. We denote this vector of variates drawn from f^{V^i} as $V^i = \{{V^i}_1, {V^i}_2, \ldots, {V^i}_{N_i^j-1}, {V^i}_{N_i^j}\}$. The auctioneer n_i sorts V^i in ascending order, creating $\bar{V}^i = \{{V^i}_{(1)}, {V^i}_{(2)}, \ldots, {V^i}_{(N_i^j-1)}, {V^i}_{(N_i^j)}\}$, keeps the top M_i bids in $\bar{V}^i$ and discards rest. We define the vector of these M_i bids as $\bar{B}^i = \{B^i_{(1)}, B^i_{(2)}, \ldots, B^i_{(M_i-1)}, B^i_{(M_i)}\}$, the vector of ranked winning bids.

We are interested in the probabilities of the various possible outcomes of the auction (see Figure 5.2). We imagine a situation where a set of bidders p_{-j} has already submitted all their bids to n_i, so that the N_i^j, M_i and m_{ji} are set (thus, N_i^j is the total number of bids that n_i will receive in addition to the m_{ji} that p_j will send, and $N_i = N_i^j + m_{ji}$). If $N_i^j > M_i$, n_i is in the scarcity state, and the bids from p_j will compete with the bids from other peers. In particular, we see that the lowest bid that p_j will have to surpass will be the $(N_i^j - M_i + 1)$-th highest bid received by n_i. In this case, the number of service

[1] In this respect, this protocol is similar to the resource allocation policy of GNUNet [149].

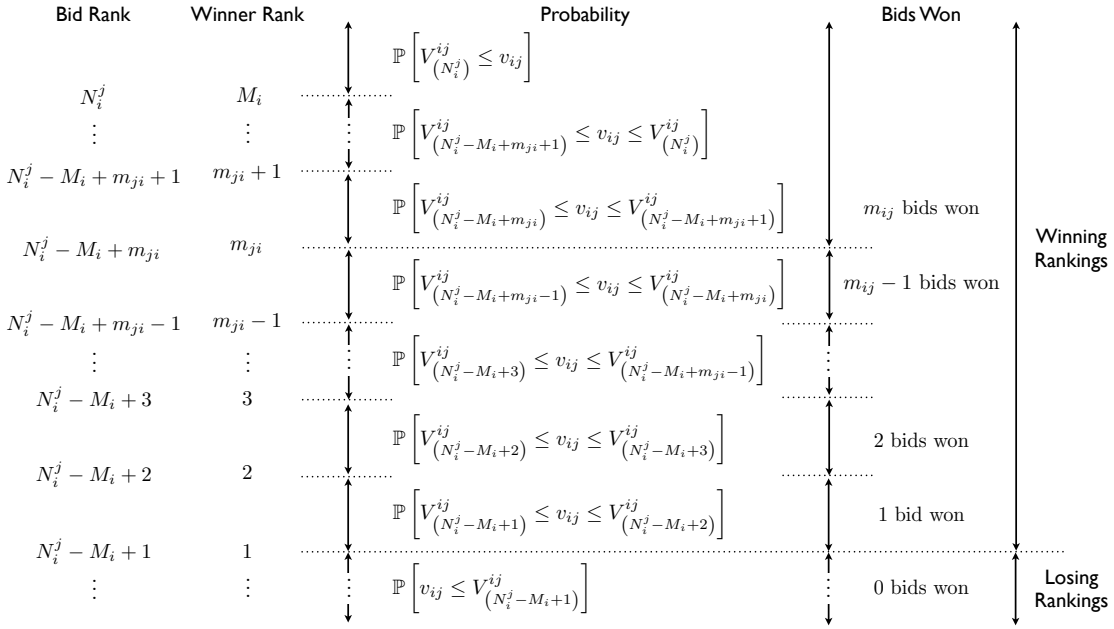


Figure 5.2: Calculating the expected number of won bids

N_i	Total number of bids that auctioneer n_i receives
N_i^j	Total number of competing bids that auctioneer n_i receives, as assessed by p_j
M_i	Total number of service units that auctioneer n_i offers
m_{ji}	Total number of bids that bidder p_j sends to auctioneer n_i
X^{ij}	Number of winning bids on auctioneer n_i that were sent by bidder p_j

Table 5.1: Auction-based swarming model notation

units that p_j can win will be bounded by the total number of units in auction M_i or the number of bids it submits, m_{ji}.

If $M_i > N_i^j$, n_i is in the overabundance state and p_j will have access to $X^{ij}_{\text{free}} = M_i - N_i^j$ contention-free service units. Clearly, the first $M_i - N_i^j$ bids from p_j will win without competition and paying zero cost. Any bids following, though, will face the same bid competition situation detailed above, with $\hat{N}_i^j = M_i - X^{ij}_{\text{free}} = N_i^j$ and $\hat{m}_{ji} = m_{ji} - X^{ij}_{\text{free}} = m_{ji} - M_i + N_i^j$ (Since by hypothesis $m_{ji} > X^{ij}_{\text{free}}$, $m'_{ji} > 0$). Then, we focus on the analysis of the scarcity state, since the overabundance state can be trivially reduced to it. We henceforth assume that $N_i^j > M_i$.

Again, we denote as X^{ij} as the number of winning bids on n_i sent by p_j. We see in Figure 5.2 that p_j will win no bids with a probability $\mathbb{P}[v_{ij} < B^i_{(1)}]$, which is equivalent to $\mathbb{P}[v_{ij} < V^i{}_{(N_i^j-M_i+1)}]$, and it will win exactly one bid with probability $\mathbb{P}[B^i_{(1)} < v_{ij} < B^i_{(2)}]$, which is equivalent to $\mathbb{P}[V^i{}_{(N_i^j-M_i+1)} < v_{ij} < V^i{}_{(N_i^j-M_i+1)}]$. This pattern continues until either all the bids that were submitted to the auctioneer have been considered (this is the situation shown in Figure 5.2) or all the service units that n_i has in auction have been considered. This bounds the maximum[2] number of bids that a bidder can win to $X^{ij}_{\max} = \min(m_{ji}, M_i)$, and this can happen with a probability of $\mathbb{P}[v_{ij} > V^i{}_{(N_i^j-M_i+X^{ij}_{\max})}] = 1 - \mathbb{P}[v_{ij} < V^i{}_{(N_i^j-M_i+X^{ij}_{\max})}]$.

Since we have that $\mathbb{P}[V^i{}_{(k)} < v_{ij} < V^i{}_{(k+1)}] = \mathbb{P}[V^i{}_{(k)} < v_{ij}] - \mathbb{P}[V^i{}_{(k+1)} < v_{ij}]$, we have that

[2] A rational peer should never send more bids than service units are available at the auctioneer, as it brings no benefit. We do not directly enforce this, however, as our protocol takes this into account automatically.

μ_{ij}, the average number of successful bids that bidder p_j will have per auction, can be found with

$$\mu_{ij} = \mathbb{E}[X^{ij}] = \sum_{k=1}^{X^{ij}_{\max}} k \left(\mathbb{P}[{V^i}_{(\varphi+k)} < v_{ij}] - \mathbb{P}[{V^i}_{(\varphi+k+1)} < v_{ij}] \right), \tag{5.1}$$

where $\varphi = N_i^j - M_i$. Since (5.1) telescopes, it can be trivially simplified to yield

$$\mu_{ij} = \sum_{k=1}^{X^{ij}_{\max}} \mathbb{P}[{V^i}_{(\varphi+k)} < v_{ij}]. \tag{5.2}$$

We find $\mathbb{P}[{V^i}_{(\varphi+k)} < v_{ij}]$ by using the $(\varphi + k)$-th *order statistics* [95, 40] of the value distribution f^{V^i}, which are formally defined as follows.

Definition 5.1. Let $V^i \geq 0$ denote a continuous random variable with probability density $f^{V^i}(v)$, and cumulative distribution $F^{V^i}(v) = \int_0^v f^{V^i}(\omega)d\omega$. Let $(V^i_{(1)}, V^i_{(2)}, \ldots, V^i_{(N_i^j)})$ denote a random sample of size N_i^j drawn on V^i, and ordered so that $V^i_{(1)} < V^i_{(2)} < \ldots < V^i_{(N_i^j)}$. Then, the $V^i_{(1)}, V^i_{(2)}, \ldots, V^i_{(N_i^j)}$ are collectively known as the *order statistics* derived from V^i. The probability density for the k-th order statistic will be denoted as $f^{V^i}_{(k)}(v)$, and can be calculated [40] as

$$f^{V^i}_{(k)}(v) = \frac{N_i^j!}{(k-1)!(N_i^j - k)!} F^{V^i}(v)^{k-1}(1 - F^{V^i}(v))^{N_i^j - k} f^{V^i}(v). \tag{5.3}$$

As is customary, we will denote the CDF of the k-th order statistic of the random variable V^i as

$$\mathbb{P}[{V^i}_{(k)} \leq v_{ij}] = F^{V^i}_{(k)}(v_{ij}) = \int_0^{v_{ij}} f^{V^i}_{(k)}(\omega)d\omega. \tag{5.4}$$

By performing the substitution $y = F^{V^i}(\omega)$ on (5.3), and assuming that $F^{V^i}(0) = 0$ we have that

$$\mathbb{P}[{V^i}_{(k)} \leq v_{ij}] = \int_0^{F^{V^i}(v_{ij})} \frac{N_i^j!}{(k-1)!(N_i^j - k)!} y^{k-1}(1-y)^{N_i^j - k} dy,$$

which can be expressed as

$$\mathbb{P}[{V^i}_{(k)} \leq v_{ij}] = \frac{1}{B(k, N_i^j - k + 1)} B(F^{V^i}(v_{ij})\,;\, k, N_i^j - k + 1), \tag{5.5}$$

where $B(k, N_i^j - k + 1)$ and $B(x\,;\, k, N_i^j - k + 1)$ are, respectively, the *beta* and *incomplete beta functions* with parameters $a = k$ and $b = N_i^j - k + 1$:

$$B(x\,;\, a, b) = \int_0^x y^{a-1}(1-y)^{b-1} dy \quad [0 \leq x \leq 1]$$
$$B(a, b) = B(1\,;\, a, b) = \int_0^1 y^{a-1}(1-y)^{b-1} dy.$$

By substituting (5.5) in (5.1), we have that

$$\mu_{ij} = \sum_{k=1}^{X^{ij}_{\max}} I(F^{V^i}(v_{ij})\,;\, \varphi + k, N_i^j - (\varphi + k) + 1), \tag{5.6}$$

where $I(x\,;\, \varphi + k, N_i^j - (\varphi + k) + 1)$ is the *regularised incomplete beta function* of x with parameters

$a = \varphi + k$ and $b = N_i^j - (\varphi + k) + 1$:

$$I(x\,;\,a,b) = \frac{B(x\,;\,a,b)}{B(a,b)} \quad [0 \leq x \leq 1].$$

Since $I(0;a,b) = 0$ and $I(1\,;\,a,b) = 1$, we have that $\lim_{v_{ij}\to 0}\mu_{ij} = 0$ and $\lim_{v_{ij}\to\infty}\mu_{ij} = X_{\max}^{ij}$, as expected: in the first case no bids are won, while in the second one the maximum number of feasible service units are won.

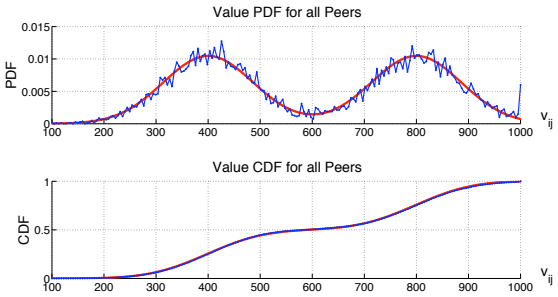


Figure 5.3: Value probability density (PDF) and cumulative distribution (CDF) for service unit valuations $f^{V^i}(v_{ij})$ and $F^{V^i}(v_{ij})$ used as input for (5.6), (5.7) and (5.8) in Figures 5.4,5.5 and 5.7. The experimental PDF and CDF are shown in blue, the analytical ones in red.

5.2.1.2 Estimating the Variance of the Number of Successful Bids

We now turn to the calculation of the second moment of X^{ij}, the number of bids that p_j wins at n_i. We have that

$$\mathbb{E}[(X^{ij})^2] = \sum_{k=1}^{X_{\max}^{ij}} k^2 \left(\mathbb{P}[{V^i}_{(\varphi+k)} < v_{ij}] - \mathbb{P}[{V^i}_{(\varphi+k+1)} < v_{ij}]\right),$$

which again trivially telescopes as

$$\mathbb{E}[(X^{ij})^2] = \sum_{k=1}^{X_{\max}^{ij}} \left(k^2 - (k-1)^2\right)\mathbb{P}[{V^i}_{(\varphi+k)} < v_{ij}] = \sum_{k=1}^{X_{\max}^{ij}} (2k-1)\mathbb{P}[{V^i}_{(\varphi+k)} < v_{ij}].$$

This implies that

$$\mathbb{E}[(X^{ij})^2] = -\mu_{ij} + 2\sum_{k=1}^{X_{\max}^{ij}} k\,\mathbb{P}[{V^i}_{(\varphi+k)} < v_{ij}],$$

which in turn means that we have for σ_{ij}^2, the variance of X^{ij}, that

$$\begin{aligned}\sigma_{ij}^2 &= -\mu_{ij}^2 - \mu_{ij} + 2\sum_{k=1}^{X_{\max}^{ij}} k\,\mathbb{P}[{V^i}_{(\varphi+k)} < v_{ij}] \\ &= -\mu_{ij}^2 - \mu_{ij} + 2\sum_{k=1}^{X_{\max}^{ij}} k\,I(F^{V^i}(v_{ij})\,;\,\varphi+k, N_i^j - (\varphi+k) + 1)\end{aligned} \tag{5.7}$$

where, again, $I(x\,;\,\varphi+k, N_i^j-(\varphi+k)+1)$ is the regularised incomplete beta function with parameters $a = \varphi + k$ and $b = N_i^j - (\varphi + k) + 1$. In this case, since we have that $\lim_{v_{ij}\to 0}\mu_{ij} = 0$ and $\lim_{v_{ij}\to\infty}\mu_{ij} = X_{\max}^{ij}$ we see that $\lim_{v_{ij}\to 0}\sigma_{ij}^2 = 0$ and $\lim_{v_{ij}\to\infty}\sigma_{ij}^2 = 0$. This is of course expected; in the first case p_j always loses all its bids, and in the second case it always wins them. There is no risk associated with either of these two cases.

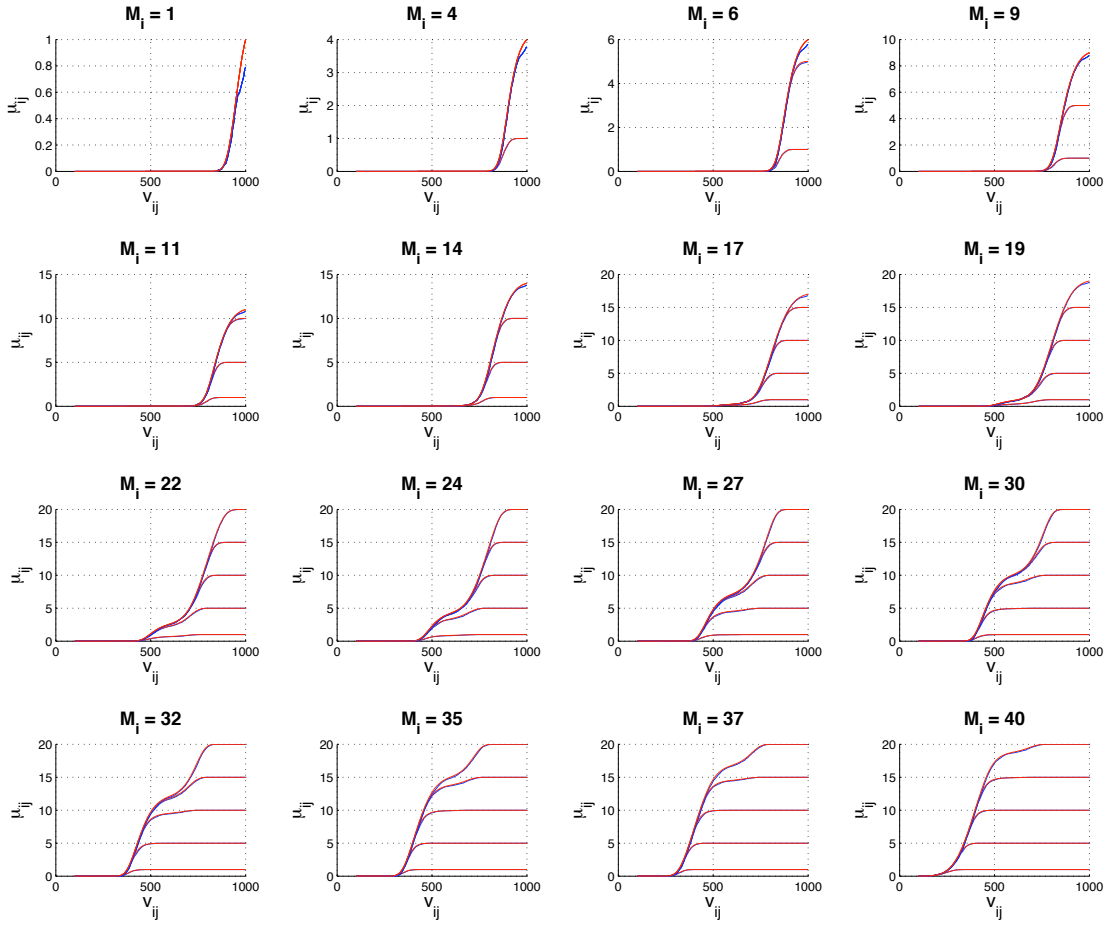


Figure 5.4: Expected number of won bids $\mathbb{E}[X^{ij}]$. Blue curves correspond to experimental simulation, red curves to (5.6). Each curve corresponds to a different $m_{ji} \in \{1, 5, 10, 15, 20\}$. All curves are present in all graphs, but overlap in the cases where $m_{ji} > M_i$ since the number of won items is clamped at M_i in that case.

5.2.1.3 Estimating the Expectation of the Utility for all Successful Bids

Finally, we calculate the expected utility of a peer. This utility, for a won bid, equals $v_{ij} - \mathsf{h}_j$ where h_j is the price that p_j needs to pay to n_i, which is found by applying the Clarke pivot rule (see Section 3.4.2.2). Then, we have for the utility of p_j that (see Figure 5.6):

$$U_{ij} = \begin{cases} v_{ij} - V_{(\varphi+1)} & \text{if} \quad V_{(\varphi+1)} < v_{ij} < V_{(\varphi+2)} & \text{(1 bid won)} \\ 2v_{ij} - V_{(\varphi+2)} - V_{(\varphi+1)} & \text{if} \quad V_{(\varphi+2)} < v_{ij} < V_{(\varphi+3)} & \text{(2 bids won)} \\ 3v_{ij} - V_{(\varphi+3)} - V_{(\varphi+2)} - V_{(\varphi+1)} & \text{if} \quad V_{(\varphi+3)} < v_{ij} < V_{(\varphi+4)} & \text{(3 bids won)} \\ 4v_{ij} - V_{(\varphi+4)} - V_{(\varphi+3)} - V_{(\varphi+2)} - V_{(\varphi+1)} & \text{if} \quad V_{(\varphi+4)} < v_{ij} < V_{(\varphi+5)} & \text{(4 bids won)} \\ & \vdots & \\ X^{ij}_{\max} v_{ij} - \sum_{n=1}^{X^{ij}_{\max}} V_{(\varphi+n)} & \text{if} \quad V_{(\varphi+X^{ij}_{\max})} < v_{ij} & \text{(All bids won)} \end{cases}$$

By a reasoning identical to that shown in Figure 5.2, we can express this utility in terms of one sided

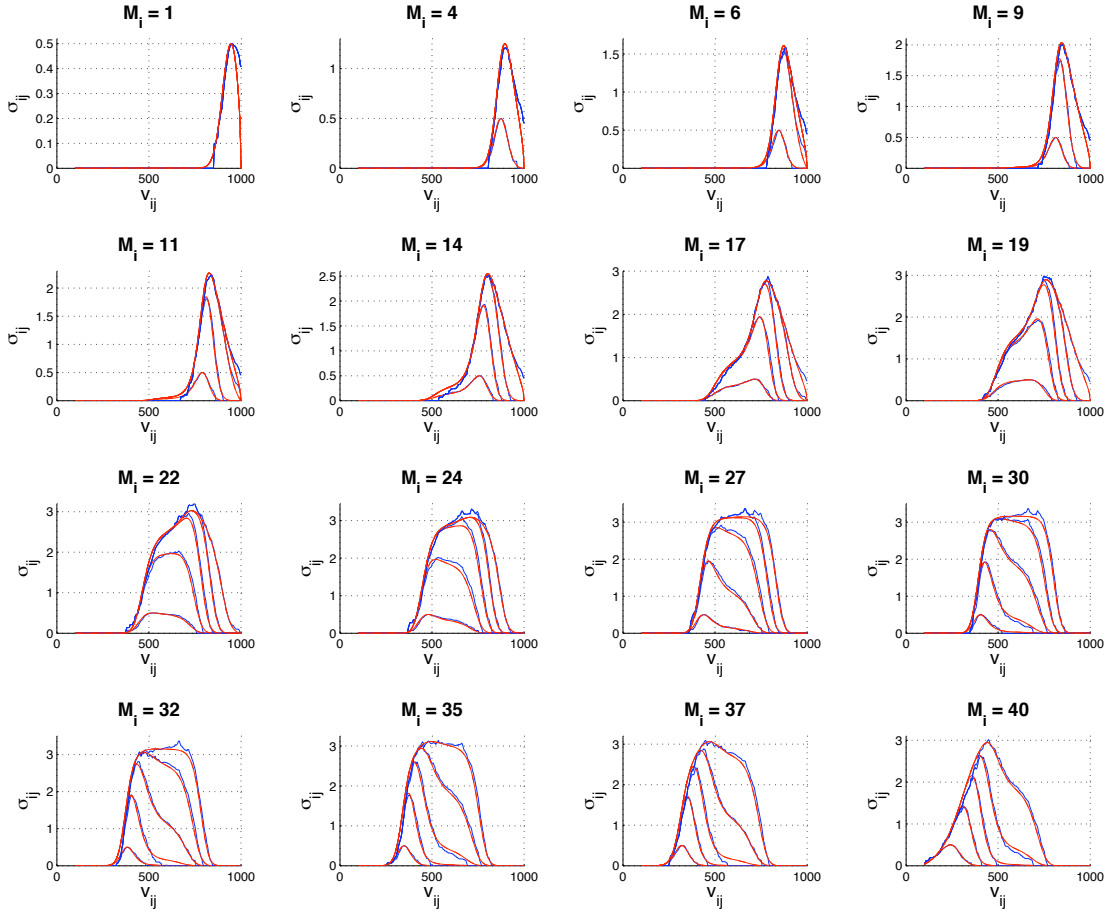


Figure 5.5: Standard deviation of the number of won bids $\left(\mathbb{E}[(X^{ij})^2] - \mathbb{E}[X^{ij}]^2\right)^{\frac{1}{2}}$. Blue curves correspond to experimental simulation, red curves to (5.7). Each curve corresponds to a different $m_{ji} \in \{1, 5, 10, 15, 20\}$. All curves are present in all graphs, but overlap in the cases where $m_{ji} > M_i$ since the number of won items is clamped at M_i in that case.

intervals in the following way (again, this is evident in Figure 5.6):

$$\begin{aligned} U_{ij} = [v_{ij} - V_{(\varphi+1)}] \quad &\text{if} \quad V_{(\varphi+1)} < v_{ij} \\ +[v_{ij} - V_{(\varphi+2)}] \quad &\text{if} \quad V_{(\varphi+2)} < v_{ij} \\ +[v_{ij} - V_{(\varphi+3)}] \quad &\text{if} \quad V_{(\varphi+3)} < v_{ij} \\ +[v_{ij} - V_{(\varphi+4)}] \quad &\text{if} \quad V_{(\varphi+4)} < v_{ij} \\ \vdots \\ +[v_{ij} - V_{\varphi+X^{ij}_{\max}}] \quad &\text{if} \quad V_{(\varphi+X^{ij}_{\max})} < v_{ij} \end{aligned}$$

and thus we have for the expected utility $\mathbb{E}[U_{ij}]$ that

$$\mathbb{E}[U_{ij}] = \sum_{k=1}^{X^{ij}_{\max}} \int_0^{v_{ij}} (v_{ij} - \omega)\, f^{V^i}_{(\varphi+k)}(\omega) d\omega,$$

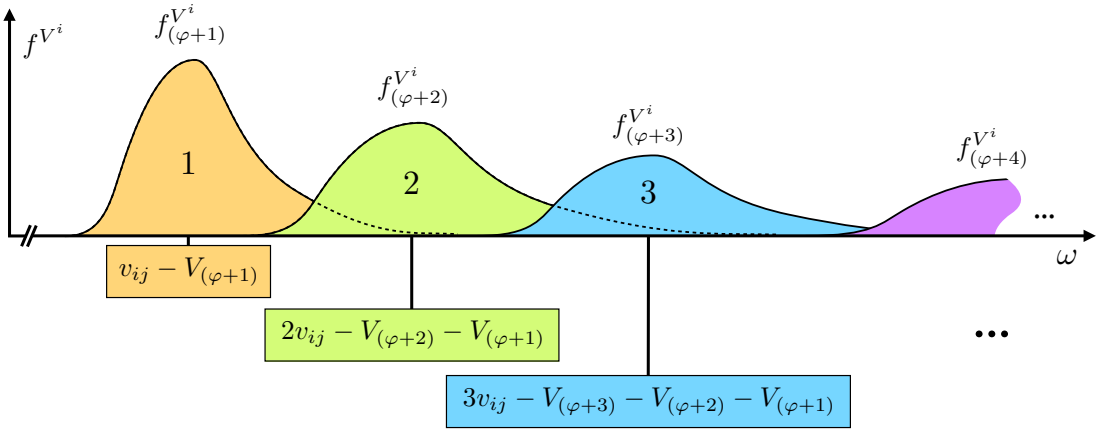


Figure 5.6: The probability density functions for the order statistics relevant in the calculation of the expected bidder utility. Each curve denotes the density function (PDF) of an order statistic (labelled in the figure itself). The coloured areas correspond to the regions over which the peer experiences a given utility, given in a box with the same background colour. The number inside each region denotes the number of bids won.

where $f^{V^i}_{(\varphi+k)}(\omega)$ is the density function (PDF) of the $(\varphi+k)$-th order statistic of V^i, as given by (5.3). If we assume that $F^{V^i}_{(\varphi+k)}(0) = 0$, the integral within the summation can be partially performed to yield

$$\int_0^{v_{ij}} (v_{ij} - \omega)\, f^{V^i}_{(\varphi+k)}(\omega) d\omega = v_{ij} F^{V^i}_{(\varphi+k)}(v_{ij}) - \int_0^{v_{ij}} \omega f^{V^i}_{(\varphi+k)}(\omega) d\omega.$$

where $F^{V^i}_{(\varphi+k)}(v_{ij})$ is just the CDF of the $(\varphi+k)$-th order statistic of V^i, as given by (5.4). The last integral in the expression above is readily solved using integration by parts, to yield

$$\int_0^{v_{ij}} \omega f^{V^i}_{(\varphi+k)}(\omega) d\omega = \omega F^{V^i}_{(\varphi+k)}(\omega)\Big|_0^{v_{ij}} - \int_0^{v_{ij}} F^{V^i}_{(\varphi+k)}(\omega) d\omega$$

Hence, we have that

$$\mu^U_{ij} = \mathbb{E}[U_{ij}] = \sum_{k=1}^{X^{ij}_{\max}} \left[v_{ij} F^{V^i}_{(\varphi+k)}(v_{ij}) - v_{ij} F^{V^i}_{(\varphi+k)}(v_{ij}) \right] + \int_0^{v_{ij}} F^{V^i}_{(\varphi+k)}(\omega) d\omega,$$

and therefore

$$\begin{aligned} \mu^U_{ij} &= \sum_{k=1}^{X^{ij}_{\max}} \int_0^{v_{ij}} F^{V^i}_{(\varphi+k)}(\omega) d\omega \\ &= \sum_{k=1}^{X^{ij}_{\max}} \int_0^{v_{ij}} I(F^{V^i}(\omega)\,;\, \varphi+k, N^j_i - (\varphi+k) + 1) d\omega. \end{aligned} \tag{5.8}$$

By exchanging summation and integration we see that

$$\mu^U_{ij} = \int_0^{v_{ij}} \mu_{ij}(\omega) d\omega,$$

where we interpret μ_{ij} as a function of the peer valuation v_{ij}, and we substitute it with the integration variable ω.

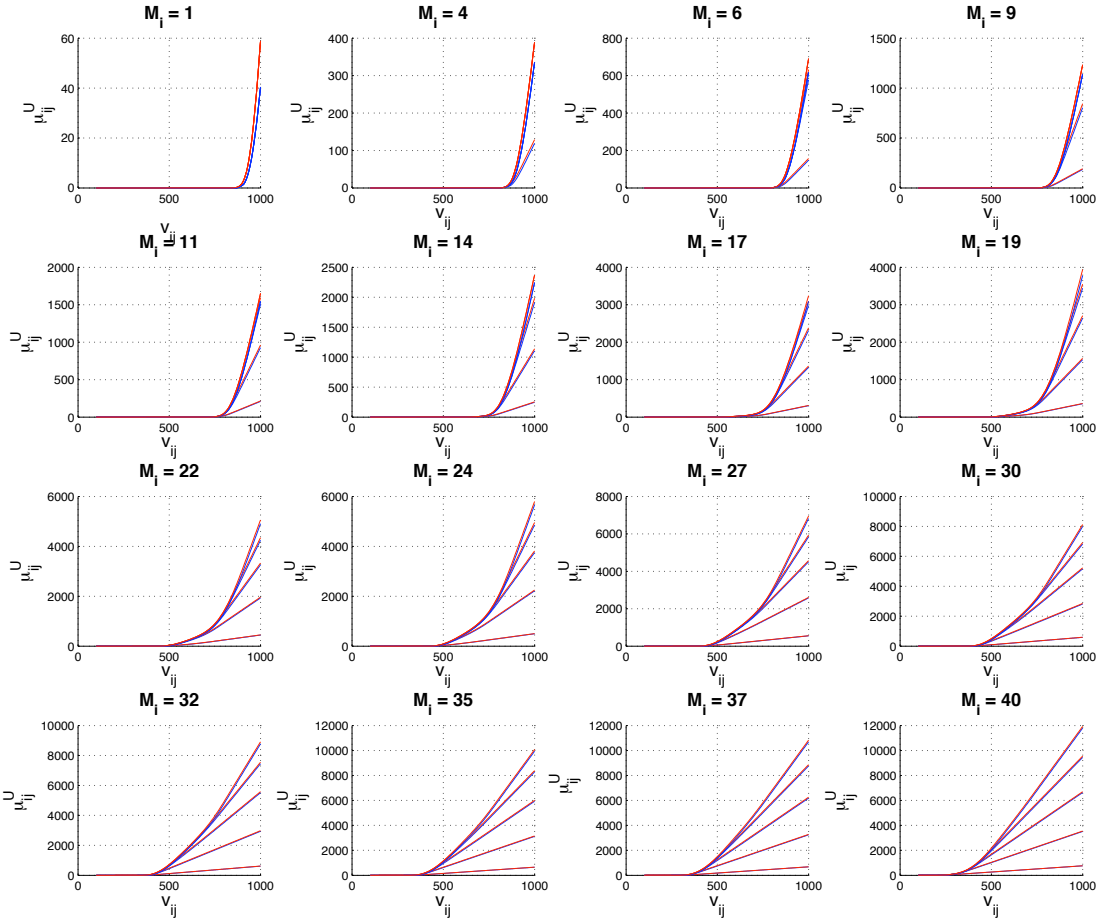


Figure 5.7: Expected utility for all won bids $\mathbb{E}[U_{ij}]$. Blue curves correspond to experimental simulation, red curves to (5.8). Each curve corresponds to a different $m_{ji} \in \{1, 5, 10, 15, 20\}$. All curves are present in all graphs, but overlap in the cases where $m_{ji} > M_i$ since the number of won items is clamped at M_i in that case.

5.2.2 Allocating Winning Bids to Service Classes

The auction procedure detailed in Section 5.2 was only used to decide whether a peer would receive service or not. Once the set of peers that will receive service is known, each one is assigned a different quality class, according to the currency reserves that the peer has with the auctioneer. We present two possibilities, one of greater accuracy but much greater computational demands, and a simpler one that nevertheless satisfies our requirements for an incentive mechanism.

For both procedures, we assume that each auctioneer can allocate its resources during the auction time window T so that each one of the winning peers p_j is assigned a service quality $q_j \in \mathcal{Q}$ so that it conforms to the structure of the w_{ji} as much as possible. We now present the first of our proposed ways of mapping contributions to quality, which achieves high accuracy at the cost of increased computational complexity.

5.2.2.1 Exact Proportional Quality Mapping

In this first approach to quality mapping, the auctioneer n_i will define a *resource* space $\Omega^i = \Omega^i_1 \times \Omega^i_2 \times \Omega^i_3 \times \ldots$ where each of the subspaces Ω^i_k corresponds to a particular resource k under the control of n_i that can be partitioned and allocated differentially to bidders (transaction delay, upload bandwidth share, etc). We assume that these allocations are *exclusive*, so that if the auctioneer allocates a bidder p_j an amount $(\omega^*_j)_k$ of resources from Ω^i_k, these will be unavailable for allocation to any other peer.

In order to relate resource allocation with service quality, we assume that the auctioneer has a *quality estimation* function $\mathsf{F}_i : \Omega^i \mapsto \mathcal{Q}$ that maps each resource allocation vector $\omega \in \Omega^i$ to a quality level $q \in \mathcal{Q}$.

We are interested in the level sets of F_i: the sets $\hat{\Omega}^i(q_c) \subset \Omega^i$ such that $\mathsf{F}_i(\omega) = q_c$ for all $\omega \in \hat{\Omega}^i(q_c)$. We thus define the correspondence $\mathsf{F}_i^{-1} : \mathcal{Q} \mapsto \Omega^i$ that gives, for each $q \in \mathcal{Q}$, its level set in Ω^i under $\mathsf{F}_i(\omega)$, and denote this as $\hat{\Omega}^i(q_c) = \mathsf{F}_i^{-1}(q_c)$. We call F_i^{-1} the *inverse quality mapping correspondence*. By definition, $\hat{\Omega}^i(q_c)$ gives all the possible resource allocations in n_i that will yield a quality of q_c. We denote $\hat{\Omega}^i(q_c)$ as the q_c*-quality level set* of peer n_i.

If Ω^i is infinitely divisible and the auctioneer has a *target aggregate quality* q_T^i that it wants to achieve per time window T, one way of achieving differentiated resource allocation for each winning bidder p_j of the M_i that will be granted service is by selecting one vector ω_j^* with $j \in \{1, 2, \ldots, M_i\}$ out of each level set $\hat{\Omega}^i(q_j)$, where q_j is the quality that will be allocated to winning bidder p_j. In this case, we propose a *proportional quality mapping* such as

$$q_j = q_T^i \frac{w_{ji}}{\sum_j w_{ji}}. \tag{5.9}$$

Then, each winning bidder p_j will be associated with a feasible quality subspace $\hat{\Omega}^i(q_j)$, so that

$$\hat{\Omega}^i(q_j) = \mathsf{F}_i^{-1}\left(q_T^i \frac{w_{ji}}{\sum_j w_{ji}}\right), \tag{5.10}$$

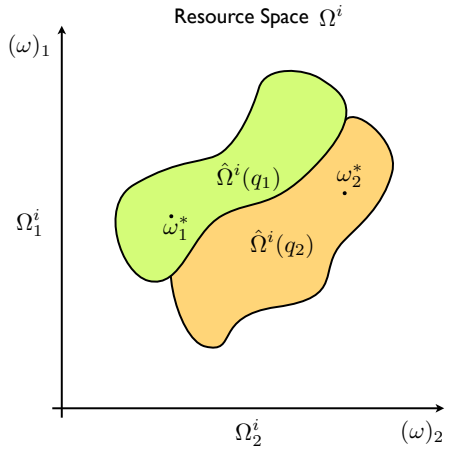
and one allocation vector $\omega_j^* \in \hat{\Omega}^i(q_j)$ will be chosen for each winning bidder p_j, subject to capacity constraints which, by our exclusivity assumption, then take the form

$$\sum_{\mathsf{p}_j \in \mathsf{P}} (\omega_j^*)_k \leq (\bar{\omega}_i)_k, \tag{5.11}$$

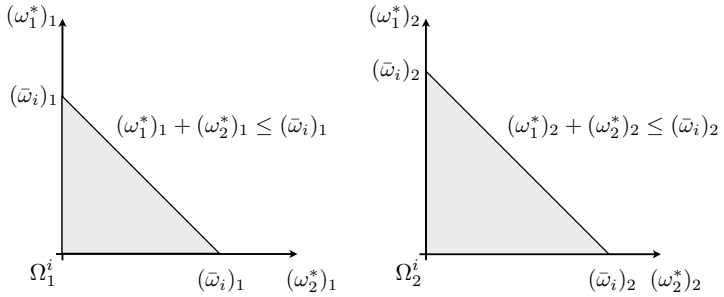
where $\bar{\omega}_i$ is the total resource vector that auctioneer n_i has, and $(\bar{\omega}_i)_k$ is the k-th component of $\bar{\omega}_i$ (the total capacity that the auctioneer n_i has for resource $k \in \Omega_k$). A conceptual example of the process just described, when applied to a resource space with only two subspaces Ω_1^i and Ω_2^i, each one of them mapped to an interval in $\mathbb{R}_{\geq 0}$ (which means that Ω^i is a finite Euclidean plane), and two peers who are to receive resource allocations is shown in Figure 5.8.

In the example in Figure 5.8, the quality levels to be associated with bidders p_1 and p_2 have been calculated as q_1 and q_2, and their feasible quality subspaces $\hat{\Omega}^i(q_1)$ and $\hat{\Omega}^i(q_2)$ defined on this basis (this is shown in Figure 5.8(a)). The proportional quality assignment problem is, in this case, to choose $\omega_1^* \in \hat{\Omega}^i(q_1)$ and $\omega_2^* \in \hat{\Omega}^i(q_2)$ such that their components lie in the shaded areas in Figure 5.8(b).

Apart from the analytic treatment (inspired by the *system of distinct representatives* technique in [169]), the main contribution of this section is that, from (5.10), the total quality q_T^i is allocated proportionally to all winning bidders so that $q_j = q_T^i \frac{w_{ji}}{\sum_j w_{ji}}$. This approach as it stands is, however, unsatisfactory for many reasons. First, the experimental calculation of the feasible quality subspaces $\hat{\Omega}^i(q_j)$ is an involved problem in and of itself. In addition, and being critically dependent on the structure of the $\hat{\Omega}^i(q_j)$, the calculation of a set of allocation vectors $\omega_j^* \in \hat{\Omega}^i(q_j)$ that, in addition, satisfy resource capacity constraints (5.11) can be a very difficult combinatorial problem, potentially with unfeasible computational demands. In summary, not only can the calculation of the ω_j^* be very computationally intensive, depending on the structure of the resource space Ω^i and the complexity of the level sets $\mathsf{F}_i^{-1}(q_j)$ and the capacity constraints (5.11), but it also requires the a-priori definition of the target quality q_T^i in



(a) Exact proportional quality mapping feasible resource subspaces (example)



(b) Exact proportional quality mapping resource capacity restrictions (example)

Figure 5.8: Example of finding the allocation vectors ω_j^* using exact proportional quality mapping

such a way that the problem is feasible[3].

5.2.2.2 Rank Quality Mapping

Even though the exact solution of 5.2.2.1 would be highly accurate, we are interested in exploring a simpler approach that can still be considered an incentive mechanism - that is, the quality $q^*(\mathsf{p}_j)$ that each bidder obtains is monotonically increasing with its account balance w_{ji}, so that peers have an incentive to continuously contribute to the peer-to-peer overlay, so that their contributions can be transferred to whichever peers they require services from and the quality they receive increases.

We propose to apply a preference relation to the set of winning peers in order to rank them, and then use their rank k to assign them to one of a set of QoS classes $\mathsf{Q}_i(k) = \{\mathsf{q}_1^i, \mathsf{q}_2^i, \ldots, \mathsf{q}_{M_i}^i\}$ that have been precomputed by the auctioneer n_i. If this preference relation orders peers according to their previous contributions, peers with greater resource donations to the overlay (as indicated by large account values w_{ji}) will obtain better quality levels.

Clearly, a very simple way of achieving this would be for the auctioneer n_i to sort the M_i winning service requests in increasing order of currency reserves, and then to assign a increasing quality service class to each request following this ordering. If we follow the technique presented in Section 5.2.1.1, we can treat these M_i account values as realisations of the random variable W^i representing the stored credit (unreciprocated contributions) of all peers towards n_i. Then, the expression equivalent to (5.1)

[3]For simple, 1-dimensional quality estimation functions this has been done in the past, one such example being [201]. The analytical framework of [201], though, is much less general than the one presented here.

can be found by considering the expected service quality of p_j,

$$\mu_{ij}^{Q} = \sum_{k=1}^{X_{\max}^{ij}} \mathsf{Q}_i(k) \left(\mathbb{P}[{W^i}_{(k)} \leq w_{ji}] - \mathbb{P}[{W^i}_{(k+1)} \leq w_{ji}] \right).$$

If we define $\mathsf{Q}_i(k)$ so that $\mathsf{Q}_i(0) = 0$, we have that

$$\mu_{ij}^{Q} = \sum_{k=1}^{X_{\max}^{ij}} \left(\mathsf{Q}_i(k) - \mathsf{Q}_i(k-1) \right) \mathbb{P}[{W^i}_{(k)} \leq w_{ji}],$$

and if we define $\Delta\mathsf{Q}_i(k) = \mathsf{Q}_i(k) - \mathsf{Q}_i(k-1)$, the expression above can be simplified to yield

$$\mu_{ij}^{Q} = \sum_{k=1}^{X_{\max}^{ij}} \Delta\mathsf{Q}_i(k) I(F^{W^i}(w_{ji})\,;\, k, M_i - k). \tag{5.12}$$

We now prove that allocating service quality as presented above is indeed an incentive mechanism, if $\mathsf{Q}_i(k)$ is increasing with the bid rank k, that is, if $\mathsf{q}_1^i < \mathsf{q}_2^i < \ldots < \mathsf{q}_{M_i}^i$.

Theorem 5.1. *Given* $\mathsf{Q}_i(k) = \{\mathsf{q}_1^i, \mathsf{q}_2^i, \ldots, \mathsf{q}_{M_i}^i\}$ *with* $\mathsf{q}_1^i < \mathsf{q}_2^i < \ldots < \mathsf{q}_{M_i}^i$*, the expected quality* μ_{ij}^{Q} *is monotonically increasing with increasing* w_{ji}*.*

Proof. First, we see that the $I(F^{W^i}(w_{ji})\,;\, k, m_{ji} - k + 1)$ are increasing in w_{ji} from the definition of $B(x; a, b)$, since it is an integral where the integrand is positive over the entire interval of integration $0 \leq x \leq 1$. This stems from the fact that $F^{W_i}(\omega) : \mathbb{R} \mapsto [0, 1]$ is non-decreasing and non-negative on its whole domain $(-\infty, \infty)$. Of course, by definition, the ranking that a peer receives is monotonically increasing with increasing w_{ji}. Since Q is increasing, we have that $\mathsf{q}_k^i > \mathsf{q}_{k-1}^i$ and $\Delta\mathsf{Q}_i(k) > 0$, and since μ_{ij}^{Q} is a linear combination with positive coefficients of increasing functions of w_{ji}, it is itself increasing with respect to the w_{ji}. □

5.3 The Bidder Process

We present a heuristic utility-maximising algorithm. Essentially, the idea is to estimate the expected utility and the number of service units that will be obtained from a given auctioneer as a function of the planned number of bids submitted to it and their value. This expected utility can be used to select the auctioneer that provides the largest expected utility for the current bid, and then submit a bid to it. The detailed specification of the peer selection heuristic for the bidder process is shown in Algorithm 2 (for variable definitions see Table 5.2).

The algorithm iterates over the service units that each peer requires. Each bidder process will send a bid for a service unit one or more times, until in expectation it receives each service unit α_k times (the obvious choice would be $\alpha_k = 1$, but $\alpha_k < 1$ could be used if the peer can cope with unavailable service units). The candidate auctioneers to which a bid might be sent include all those peers which can provide the service unit in question and to whose auctions the bidder has not submitted a bid yet. For each one of these candidate peers, the bidder will calculate ΔU_{ij}, the increase in its utility from sending each one additional bid beyond those that are already scheduled for that peer (the $\mu_{ij}(n_i^*)$ and $U_{ij}(v_{ji}, m_{ji}, N_i^j)$ in Algorithm 2 are just (5.6) and (5.8) respectively).

The peer then compares, between all candidate auctioneers, the additional expected utility that it will experience as a result of an additional bid, and the peer for which this magnitude is maximal is selected.

Algorithm 2 Expected utility maximisation heuristic for p_j

$m_{ji} = 0 \, \forall n_i$
for all service units s_k **do**
  {*Initialise bidder process state*}
  $\mathbf{R}_k^j = \emptyset$
  $\mathbf{N}_k^j = 0$
  while $\mathbf{N}_k^j < \alpha_k$ **do**
    {*Obtain candidate auctioneer set*}
    $A_k^j = \mathbf{P}_k \setminus \mathbf{R}_k^j$
    {*Maximise the value of the next bid value the candidate auctioneers*}
    $\Delta U_{ij} = U_{ij}(v_{ji}, m_{ji} + 1, N_i^j) - U_{ij}(v_{ji}, m_{ji}, N_i^j)$
    $\Delta \mu_{ij} = \mu_{ij}(v_{ji}, m_{ji} + 1, N_i^j) - \mu_{ij}(v_{ji}, m_{ji}, N_i^j)$
    $n_i^* = \arg\max_{n_i \in A_k^j} \Delta U_{ij}$
    $\Delta \mu_{ij}^* = \Delta \mu_{ij}(n_i^*)$
    Send a bid of value v_{ij} to peer n^*
    {*Update bidder process state*}
    $\mathbf{R}_k^j(\mathsf{p}_j) = \mathbf{R}_k^j(\mathsf{p}_j) \cup n_i^*$
    $\mathbf{N}_k^j = \mathbf{N}_k^j + \Delta \mu_{ij}^*$
    $m_{ji}(n_i^*) = m_{ji}(n_i^*) + 1$;
  end while
end for

m_{ji}	Number of bids for n_i that p_j has scheduled
$\mathbf{R}_k^j$	List of peers to which p_j has scheduled a bid for service unit s_k
$\mathbf{N}_k^j$	Expected number of times that p_j will receive service unit s_k
α_k	Threshold value for the expected number of times that service unit s_k will be received, $0 < \alpha_k \leq 1$.
A_k^j	List of peers that can provide service unit s_k to p_j, and which have not been scheduled to receive a bid yet
$\mathbf{P}_k$	List of all peers that can provide service unit s_k
$\Delta U_{ij}(m_{ji})$	Expected additional utility that p_j would obtain by bidding $m_{ji} + 1$ times instead of just m_{ji} times with auctioneer n_i
$\Delta \mu_{ij}(m_{ji})$	Expected number of additional service units that p_j would obtain by bidding $m_{ji} + 1$ times instead of just m_{ji} times with auctioneer n_i
N_i^j	Number bids per round that compete in n_i with those sent by p_j

Table 5.2: Peer selection algorithm notation

In order to execute Algorithm 2, the bidder will require, from the market system, some information regarding all tentative auctioneers. In particular, p_j will require $F^{V^i}(v_{ij})$, the CDF of valuations at the auctioneer, M_i, the number of service units being auctioned in the current round, N_i, the total number of bids that the auctioneer receives per round, and $\mathbf{P}(s_k)$, the list of all peers that can provide service unit s_k for all units s_k that p_j requires. The only processing that p_j will perform before performing Algorithm 2 will be to obtain N_i^j by removing from N_i its own bids to n_i; the rest of this information will be used as given (the value v_{ij} is privately calculated by p_j as detailed in Section 5.3.1). We shall not address the strategic issues surrounding the truthful revelation of these values.

5.3.1 The Peer Valuating Function

We assume that each peer will have access to a vector ζ_i of network measurements and peer performance indicators[4], either directly measured or obtained through the market system $\mathcal{M}$. Since we require these measurements to be as reusable between peers as possible, we impose the requirement that they must be

[4] Although each peer might choose to keep interaction logs of its transactions with other peers to better estimate their value, this is not a requirement.

defined so that they explicitly exclude the effect of the quality allocation procedure detailed in Section 5.2.2.

The bidder has a value function $v_{ij} = v_j(\zeta_{ij}, \mu_{ij}^Q)$ where the vector ζ_{ij} captures not only the private preferences of each peer and the reported quality indicators that each auctioneer might provide, but also the possible effects that the network will have over the service, independently of the actions of the auctioneer[5].

The μ_{ij}^Q used as input for the value equation above is the expected quality that the server will assign to the bidder, and it is calculated using (5.12). This means that the value v_{ij} that a peer p_j assigns to a given auctioneer n_i is a function not only of the overlay network path properties ζ_{ij} between them, but also of the expected quality class to which it would be mapped in n_i, which is in itself a function of its past contributions. This gives us our incentives mechanism property: peers will find that auctioneers provide higher quality service as their contributions to them increase, when compared to the contributions of the other peers competing with them.

The calculation of v_{ij} proceeds as follows. First, p_j estimates the expected quality μ_{ij}^Q that it will receive from an auctioneer n_i. This is achieved by querying the market system $\mathcal{M}$ for $F^{W^i}(w_{ji})$, the CDF of the currency accounts in n_i, and also for its quality mapping function $\mathsf{Q}_i(k)$. Using these and its own w_{ji} value on (5.12), the prospective bidder can estimate μ_{ij}^Q, and with that, $v_{ij} = v_j(\zeta_{ij}, \mu_{ij}^Q)$. Since p_j requires v_{ij} for all its prospective auctioneers, this value calculation is performed once for each one of them. Again, an analysis of the incentives for the truthful revelation of these values will be out of scope for this thesis.

5.3.2 Alternative Peer Selection Functions

Even though we presented a particular optimisation heuristic for peer selection relying on (5.7), the framework presented can be trivially adapted to use other functions. Direct examples would include the minimisation of the standard deviation of the number of bids won (5.7) with a minimum utility constraint, or the maximisation of any increasing function of the expected bid set utility (5.8). In both of these cases (or indeed any other), the idea would be to calculate the added benefit or degradation that a new bid to a given auctioneer brings to the bidder, and schedule a bid for the auctioneer that optimises, for that bid, whatever the system designer has chosen the peer selection function to be.

5.4 The Accounting Process

In this case, the accounting process is reduced to *PledgeRoute* operating as a distributed currency platform (see Chapter 4, Section 4.9). The objective of the accounting process is only to keep track of the payments given and received, so that μ_{ij} and μ_{ij}^Q can be estimated accurately, performing any currency transfers necessary to pay for services and to ensure adequate service quality from other peers. In this case, we shall distinguish two kinds of bids: *buy requests* and *withdrawal orders*. Bid requests are, essentially, altruistic requests, and may be immediately discarded if the maximum credit for the peer is insufficient to cover for them. Withdrawal orders, on the other hand, being reciprocative in nature, will always be considered for competition in auctions. In this case, though, since we are concerned not only with incentives but also with overlay optimisation, there is no guarantee that having previous contributions with the auctioneer is enough to be granted service.

5.5 Example: Chunk Swarming for Real-Time Streaming

As a motivating example, we apply the presented framework to a specific swarming protocol problem: chunk swarming for real-time streaming using a *mesh/pull* protocol (see Section 2.2).

[5] We shall not make a distinction between these two kinds of information regarding incentives for the time being, though - the reader is referred to Chapter 6 for a deeper analysis of this issue.

A peer (the *peercaster*, p_p) takes a streaming source such as video and segments it to produce stream data units (*chunks*) at a constant *stream rate* of R chunks per interval T. Peers query the market system $\mathcal{M}$ to learn which chunks other peers have, and they select a subset of these peers from which to request chunks.

The two most important performance objectives for a peer-to-peer streaming system of this kind are *lag* and *continuity*:

- *Stream lag*: The delay between a chunk being generated by the peercaster n_p and the same chunk being received by a given downstream consumer peer. Low stream lag means that the stream is closer to real time, and thus peers capable of delivering lower lag chunks will have higher value.
- *Stream continuity*: The percentage of stream data units that arrived on time to be useful to a downstream consumer peer. High stream continuity enables minimal distortion in stream reconstruction. In our case, we leave peers free to set their continuity at whatever level they choose.

The basic assumptions for this model are then that service units are identified with *chunks*, that peers know their valuation function v_{ij}, and that each peer n_i will drive its resource allocation using the Vickrey auction model presented in the previous sections. Every peer on its bidder role with send m_{ji} bids to each auctioneer whose content it is interested in, where m_{ji} will be calculated using a procedure like that described in Section 5.3: peers will estimate the expected value per bid that each one the candidate auctioneers can provide, which will take into account the quality that they expect to receive as a function of their past contributions.

In summary, during each time window of duration T, each chunk auctioneer n_i receives N_i bids for the M_i chunks which it will upload in the next time window. The top M_i bids will be served, and each bidder p_j that wins $X^{ij} > 0$ bids will be charged an amount equal to the sum of the values of the top X^{ij} losing bids. Then, each of the winning bids will be assigned a service time inversely proportional to their account balances w_{ji}, so that the bids of peers with higher account values will be serviced first. Since chunk values will be inversely proportional to delay, only peers that contribute their upload capacity to the overlay will be able to gain access to the lowest lag chunks.

5.5.1 Chunk Swarming: The Peer Valuation Function

The basic engineering objective of the proposed incentive mechanism will be the minimisation of *stream lag*. Therefore, we require a formal definition of stream lag that can be used to construct the valuation function v_{ij} and the rank quality mapping $\mathsf{Q}_i(k)$.

As discussed throughout Chapter 2, synthetic coordinates provide efficient network locality information while reducing complexity at the application layer. In this case, we rely on situating peers in an abstract *delay space*, and then relating distance over this space to network layer delays through the use of a *metric* d, so that the delay between n_i and n_j will be denoted as $d(n_i, n_j)$. Thus, we associate a coordinate to each peer so that the d-distance between their coordinates will be a good approximation to the network latency between the peers. From now on we will refer as *delay* to this synthetic-coordinate derived distance.

Since we are assuming no synergy effects, all chunks from a given auctioneer should have the same value, and thus, the same lag. This will not be possible in the strict sense, since we will assume that chunks are not sent concurrently, but one after another at the upload speed of the auctioneer. To approximate the no-synergy condition as much as possible, however, auctioneers will upload all chunks won by a given peer immediately after one another, so that the time difference between the transmission of each chunk is minimised. The consideration of cases with synergy requires the use of a full-fledged combinatorial Vickrey auction [91], possibly along with the definition of a suitable bidding language. We consider this case beyond the scope of the present thesis.

$L(c_n, n_k)$	Chunk lag of c_n at peer n_k
$t_Q(n_j, n_k)$	Time from the start of the service interval T that will elapse before n_j finishes its chunk transfer to n_k
$d(n_j, n_k)$	Synthetic coordinate delay from n_j to n_k

Table 5.3: Components of Chunk Lag notation

Definition 5.2 (Continuous Chunk Lag)**.** We define the *chunk lag* for a chunk c_n at a given peer n_j as the amount of time that has passed since c_n was produced at the peercaster n_p and the time in which it is received by n_j. This implies that the minimum chunk lag between two peers will be equal to the delay between them. We calculate the chunk lag L_c at n_k of a chunk c_n downloaded from n_j as the chunk lag of c_n at n_j (assumed to be truthfully revealed by the sender - we defer the consideration of truthful revelation of QoS parameters, such as chunk lag, to our discussion on the principal-agent model in Chapter 6), plus the time t_Q that elapses before n_k finishes sending the chunk, plus the network delay from n_j to n_k (see Table 5.3). This model is similar to the one we present in Figure 6.2, and can be formalised as

$$L_c(c_n, n_k) = \begin{cases} 0 & \text{if } n_k \text{ is the peercaster } n_p \\ L(c_n, n_j) + t_Q(n_j, n_k) + d(n_j, n_k) & \text{if } n_k \text{ downloads } c_n \text{ from } n_j. \end{cases} \tag{5.13}$$

Definition 5.3 (Continuous Peer Lag)**.** We define the *peer lag* $L_p(n_j, k)$ of n_j as the average chunk lag of the last k **contiguous** chunks that it has received. Of course, $L_p(n_j, k) \geq L_c(c_n, n_k)$ for all chunks c_n in n_j.

Although each peer can potentially have its own valuation for each chunk, it is clear that, in general, v_{ij} will be reduced for large values of $L_c(c_n, n_k)$, since we assumed that peers attempt to minimise their stream lag. Thus, the peer valuation function $v_{ij}(L_p(n_j, k))$ is a decreasing function of the peer lag L_p.

5.5.2 Chunk Swarming: The Auctioneer Process

Another important decision in terms of mapping the abstract model presented in Section 5.2 to a specific example (in addition to the consideration of v_{ij} as a decreasing function of L_k) is the determination of the M_i rank quality classes in such a way that increasing QoS from the point of view of the auctioneer translates into higher value from the point of view of the bidder. We will define the resource space Ω^i of the bidder as the time window of duration T during which it can upload chunks. We assume that T is just enough to upload M_i chunks, so that T can then be partitioned into M_i disjoint upload *chunk slots*.

Once the set of winning peers has been decided by taking the top M_i received bids, the auctioneer will decide the assignment of chunk slots (quality classes) to winning bids (chunks) by *sorting the list of winning bids according to the contribution account balances of their corresponding peers, and then proceeding with chunk uploads by following this list*. Thus, account balances decide the order on which the bidders will be served, which itself defines the delay $t_Q(n_j, n_k)$ for chunk transmission completion. Clearly, $t_Q(n_j, n_k)$ will be a monotonically decreasing function of the account balances w_{ji}, as the more unreciprocated contributions a peer has, the sooner its chunks are sent. Clearly, in this case $\mathsf{Q}_i(k)$ maps increasing contribution account balance to reduced lag by scheduling bids to chunk slots through the ranking technique presented in Section 5.2.2.

The previous discussion is predicated on the assumption that an upload slot can only be assigned to a single peer, which is the case for network layer unicast. The consideration of multicast scenarios is complicated by the fact that different members of a multicast group could have provided very different contributions to the overlay, making reciprocation to the entire multicast tree an ill-defined concept. We shall consider network layer multicast out of the scope of the present thesis.

5.5.3 Chunk Swarming: The Quality-Resource Allocation Function

We now estimate the expected number of *chunk slots* that the bid of a peer p_j will have to wait before finishing transmission, as a function of its past contributions w_{ji}. We see that the chunk will finish transmission after 1 slot with probability $\mathbb{P}[W^i_{(M_i-1)} < w_{ji}]$, if it is at the top of the contribution account ranking (above the rest $M_i - 1$ bids). It will finish after 2 slots with probability $\mathbb{P}[W^i_{(M_i-2)} < w_{ji} < W^i_{(M_i-1)}] = \mathbb{P}[W^i_{(M_i-2)} < w_{ji}] - \mathbb{P}[W^i_{(M_i-1)} < w_{ji}]$, after $(M_i - 1)$ slots with probability $\mathbb{P}[W^i_{(1)} < w_{ji} < W^i_{(2)}] = \mathbb{P}[W^i_{(1)} < w_{ji}] - \mathbb{P}[W^i_{(2)} < w_{ji}]$ and after M_i slots with probability $\mathbb{P}[w_{ji} < W^i_{(1)}] = 1 - \mathbb{P}[W^i_{(1)} < w_{ji}]$. Then, by following a procedure identical to that one in Section 5.2.1.1, we see that the expected number of chunk slots $\mathcal{D}_{ij}$ that a request will require for service is given by

$$\mathcal{D}_{ij} = M_i - \sum_{k=1}^{M_i-1} I(F^{W^i}(w_{ji}); k, M_i - k). \tag{5.14}$$

Of course, we have that $\lim_{w_{ji}\to 0} \mathcal{D}_{ij} = M_i$ and $\lim_{w_{ji}\to\infty} \mathcal{D}_{ij} = 1$, as expected. Since we have that

$$t_Q(n_i, n_j) = \frac{T}{M_i}\mathcal{D}_{ij},$$

we focus our analysis on $\mathcal{D}_{ij}$.

Theorem 5.2. *The ranked chunk slot function $\mathcal{D}_{ij}$ is monotonically decreasing with increasing w_{ji}.*

Proof. Since F^{W^i} is a CDF, it is increasing with w_{ji}. Furthermore, since a sum of increasing functions is increasing itself, we only need to prove that $I(x; k, M - k - 1)$ is increasing to prove that $\mathcal{D}_{ij}$ is decreasing. This follows from the definition of $B(x; a, b)$, since it is an integral where the integrand is never negative in the domain $0 \leq x \leq 1$, which is the range of F^{W^i}. It follows that $\mathcal{D}_{ij}(w_{ji})$ is monotonically decreasing. □

5.5.4 Chunk Swarming: The Bidder Process

We now describe the operation of the bidder process, which can be found in greater detail in Algorithm 3. In every bidding round of length T, the peercaster generates a new chunk. Peers will get the bidding and account cumulative distributions (F^{V^i} and F^{W^i}) from their potential auctioneers n_i (or the market system $\mathcal{M}$), as well as their peer lag $L_p(n_i, k)$, their upload capacity M_i in chunks per bidding interval and their total bid load level N_i. By subtracting from N_i the load that they themselves impose on their prospective auctioneers (the bids that they have sent to them in the recent past), each peer p_j can estimate N^j_i, the number of competing bids they will face in n_i. Using w_{ji}, f^{W^i} and M_i, peers estimate the expected service delay D_{ij}, and use it along with their overlay link delay $d(n_i, n_j)$ to calculate the total expected peer lag from p_p if they download from each prospective auctioneer n_i as

$$L_p(n_j, k) = L_p(n_i, k) + t_Q(n_i, n_j) + d(n_i, n_j).$$

Then, using their value mapping function, peers can calculate their candidate auctioneer valuations $v_{ij}(L_p(n_j, k))$, which then can be used alongside f^{V^i}, M_i and N^j_i to calculate their expected increase both in the number of winning bids $\Delta\mu_{ij}$ and their utility ΔU_{ij}. These, in turn, can be used to drive the peer selection procedure (Algorithm 2) in order to obtain the number m_{ji} of bids that each bidder p_j will send to each auctioneer n_i.

5.5.5 Chunk Swarming: Simulation Experiments

Since both swarming protocols and incentive mechanisms are essentially resource allocation mechanisms with possibly conflicting goals, the design of systems that achieve both can be very challenging. As a

$\mathbf{c}_i$	Set of all chunks that peer n_i can upload, either because it has already downloaded them, or created them (in the case of the peercaster).
$\mathbf{C}^t$	Set of all chunks that are available for download at time t
$\mathbf{C}^t_{-j}$	Set of all chunks that peer n_j has yet to download
$m_{ji}(n_i^*)$	Number of bids for auctioneer n_i that p_j has scheduled
$\mathbf{R}^j_k$	Set of candidate auctioneers n_i to which bidder p_j has already scheduled a bid for c_k
$\mathbf{N}^j_k$	Number of times that bidder p_j expects to receive chunk c_k
$\mathbf{P}_k$	Set of all potential auctioneers n_i that have chunk c_k
A^j_k	Set of candidate auctioneers that have chunk c_k, but have not been scheduled by p_j for a bid yet
$\Delta U_{ij}(m_{ji})$	Expected additional utility that p_j would obtain by bidding $m_{ji}+1$ times instead of just m_{ji} times with auctioneer n_i
$\Delta\mu_{ij}(m_{ji})$	Expected number of additional chunks that p_j would obtain by bidding $m_{ji}+1$ times instead of just m_{ji} times with auctioneer n_i

Table 5.4: Chunk swarming bidder process notation

Algorithm 3 Bidding behaviour for all peers

Require: v_{ij}, $f^V(n_i)$
  for all $t : 1 < t < t_{\max}$ **do**
    $\mathbf{c}_p = \mathbf{c}_p \cup c_t$ {*Every round, the peercaster generates a new chunk*}
    $\mathbf{C}^t = \bigcup_{n_j \in \mathcal{G}} \mathbf{c}_j$
    for all $n_j \in \mathcal{G}$ (in random order) **do**
      {*Iterate over all peers* n_j}
      $\mathbf{C}^t_{-j} = \mathbf{C}^t \setminus \mathbf{c}_j$ {n_j *extracts the set of chunks that it has not yet downloaded*}
      $m_{ji} = 0 \quad \forall n_i \in \mathcal{G}$ {*Initialise* m_{ji}}
      for all $c_k \in \mathbf{C}^t_{-j}$ **do**
        {*Iterate over the interesting chunks for* n_j}
        $\mathbf{R}^j_k = \emptyset$ {n_j *initialises* $\mathbf{R}^j_k$}
        $\mathbf{N}^j_k = 0$ {n_j *initialises* $\mathbf{N}^j_k$}
        $\mathbf{P}^j_k =$ set of all potential auctioneers $n_i \in \mathcal{G}$ that have chunk c_k
        while $\mathbf{N}(c_k) < 1$ **do**
          {*Iterate until, in expectation,* c_k *is received once*}
          $A^j_k = \mathbf{P}_k \setminus \mathbf{R}^j_k$ {*Set of peers that have* c_k*, but are not scheduled for a bid yet*}
          for all $n_i \in A^j_k$ **do**
            {*Iterate over candidate auctioneers*}
            Calculate $\Delta\mu_{ij}(m_{ji})$ using current m_{ji}
            Calculate $\Delta U_{ij}(m_{ji})$ using current m_{ji}
          end for
          {*Select auctioneer* n_i^* *for which* $\Delta U_{ji}(m_{ji})$ *is maximal*}
          $n_i^* = \arg\max_{n_i \in A^j_k} \Delta U_{ij}(m_{ji}, n_i)$
          $\Delta\mu^*_{ij} = \Delta\mu_{ij}(m_{ji}, n_i^*)$
          Send a bid of value v_{ij} to peer n_i^*
          $\mathbf{R}^j_k = \mathbf{R}^j_k \cup n_i^*$ {*Add* n_i^* *to the list of peers with scheduled bids for* c_k}
          $\mathbf{N}^j_k = \mathbf{N}^j_k + \Delta\mu^*_{ij}$ {*Add* $\Delta\mu^*_{ij}$ *to the expected number of times that* n_i *will receive* c_k}
          $m_{ji}(n_i^*) = m_{ji}(n_i^*) + 1$ {*Increment the total number of scheduled bids for peer* n_i^*}
        end while
      end for
    end for
  end for

consequence, the clean delineation of which protocol elements have swarm construction or incentives mechanism goals is of interest.

In the case of the chunk swarming scheme just presented, *chunk auctions* are the main tool to balance stream lag optimisation with peer capacity constraints, and *rank quality mapping* is the main incentive mechanism tool. The value function v_{ij} brings these two elements together, in such a way that different peers can have different tradeoffs between swarm overlay optimisation and the quality differentiation provided by the incentives mechanism.

In this case, however, the swarm optimisation element is much more complex than the incentive mechanism. This was done as an explicit design choice, so that the characterisation of the incentive mechanism could be essentially completed in Section 5.2.2. The swarm building characteristics of the system, however, have not been addressed and their complexity precludes a casual mathematical formalism. We now turn to a simulation-based analysis of the swarming subsystem.

5.5.5.1 Peer Distribution in Delay Space

Although synthetic coordinates can be used for arbitrary metric spaces, in this case and for ease of presentation, a finite, two-dimensional Euclidean plane is used; we call it the *delay plane*. This means that peers are created with a position in delay space that is a vector $(x, y) \in \mathbb{R}$ such that $0 < x < 100$ and $0 < y < 100$, and the delay between peers is the Euclidean distance between their coordinates. The distribution of the peers in delay space is varied to achieve different levels of clustering.

We present three different peer placement configurations in delay space. The first one is a simple *uniform* distribution, where the coordinates of each peer are randomly drawn from a two dimensional uniform distribution. Two cases of interest are explored in this case: a peercaster artificially placed at the mean of the distribution (at the middle of the delay plane), and a peercaster asymmetrically placed to one of the sides of the delay space. The second synthetic delay configuration is a *clustered* distribution based on the superposition of uniform distributions. Essentially, five different two dimensional uniform distributions are created: four of them lie predominantly in one of the quadrants of the delay plane, and the fifth one spans the entire delay plane. Each of these five densities is used one fifth of the time when generating a new peer. The final delay configuration is obtained by embedding a small subset of the *King* dataset to the Euclidean plane. This is done using a simple quadratic error minimisation algorithm roughly based on Vivaldi [90, 94]. The embedding to a two dimensional Euclidean surface is, of course, of limited accuracy, but this does not affect the specifics of the algorithm to be tested.

5.5.5.2 System Performance Measures

In order to assess the performance of our example protocol in the different delay distribution scenarios presented in Section 5.5.5.1, we shall make use of the following measures.

- The **Discrete Peer Lag** corresponds to the number of chunks between the latest chunk produced by the peercaster and the latest chunk that a peer has for which the previous 10 chunks were received successfully. Thus, it is a measure of how long does it take for a peer to acquire enough chunks so that the stream can be considered stable, and measures the distribution efficiency of the overlay. It is defined as

  $$L_p(n_i) = c_p - c_k,$$

  where c_p is the index of the latest chunk produced by the peercaster, and c_k is the index of the latest chunk on n_i such that $\{c_{k-10}, c_{k-9}, \ldots, c_k\}$ have all been correctly received by n_i. We call this set of 10 chunks used for the determination of the peer discrete lag the *lag interval* of the peer.

- The **Stream Lag** corresponds to the minimum delay required for a chunk to get to the receiving peer through the overlay, disregarding processing time at each peer. It is defined as the sum of

the delays of the overlay links required to deliver the chunk to n_i through the overlay. We can formalise this if we define P_{pi} as the overlay route taken by the chunk from the peercaster to the receiving peer n_i. Then, we have that

$$L_s(n_i) = \sum_{(n_j,n_k)\in P_{pi}} d(n_j, n_k).$$

- The **Stream Lag Stretch** corresponds to the ratio between the time required to send a chunk through the overlay, disregarding processing time at each peer, and the time it would take for the chunk to arrive to the receiving peer directly from the peercaster (a *straight line* in delay space). Therefore, it is a purely topological measure of the efficiency of the distribution paths taken by the chunks. It is defined as the sum of the delays of the overlay links required to deliver the chunk to n_i through the overlay and the delay of the direct overlay link (n_p, n_i) from the peercaster to the target peer. Formally, we have that

$$C(n_i) = \frac{1}{d(n_p, n_i)} \sum_{(n_j,n_k)\in P_{pi}} d(n_j, n_k) = \frac{L_s(n_i)}{d(n_p, n_i)}.$$

- The **Leading Edge Continuity** is the ratio of the number of chunks that the peer has successfully received from the start of its lag interval to its newest chunk, and the total number of chunks in this interval, including those that the peer has not received yet. It is a measure of variability in the acquisition of chunks, and may be used to detect mismatches between the stream rate and the effective rate at which the overlay can deliver the stream.

- The **Peer Currency Balance** is simply the difference between the total peer income and its total cost, as a result from buying and selling chunks in auctions. It is a measure of the role of a peer as either consumer of resources or provider of resources.

- The **Normalised Lost Bid Rate**: This is the ratio between the total number of losing bids (those that did not win a chunk) and the total number of peers in the system. It is a measure of how busy a peer is.

- The **Protocol Performance Measure** is a synthetic measure of how well the protocol behaved in a given delay distribution. It is calculated as

$$\mathsf{m}(n_i) = \frac{L_p(n_i)\, C(n_i)}{N_T}, \tag{5.15}$$

where $L_p(n_i)$ and $C(n_i)$ are the peer lag and the leading edge continuity as defined above, and N_T corresponds to the number of chunks that the peercaster has generated. Since $0 < C(n_i) < 1$ and $0 < L_c(n_i) < N_T$, we have that $0 < \mathsf{m}(n_i) < 1$.

5.5.5.3 Protocol Behaviour in Different Peer Delay Distributions

At the start of the simulation run, the system generates a peer delay distribution as reported in Section 5.5.5.1 and distinguishes one peer as the peercaster. This peer is removed from its position in delay space and placed either at the centre of the delay plane, or at one side. This simple change has a profound impact on protocol performance, as shown by the representative delay configurations (and their induced traffic distribution dynamics) found on Figure 5.5.5.3.

The system defines a stream rate of 1 chunk per round, and a limit of 5 bids per peer for any single chunk. The value distribution f^{V^i} is implemented with histograms with 30 bins over the domain of

possible values. Upload capacity is expressed in chunks per round, and it is generated using a uniform distribution over the integers $n \in [3, 9]$ with a mean of 6 chunks per round (we consider such over-provisioned peers because, as shown below, the algorithm that we discuss generates very unequally distributed loads among overlay peers, with a small number of peers carrying most of the load).

Simulation then proceeds by episodes called *rounds*. Each simulation round commences with each peer calculating its stream lag, chunk lag and border continuity, since they will be needed for the evaluation of the valuation function v_{ij}. In this case, the valuation function is a linear, decreasing function of the lag such as

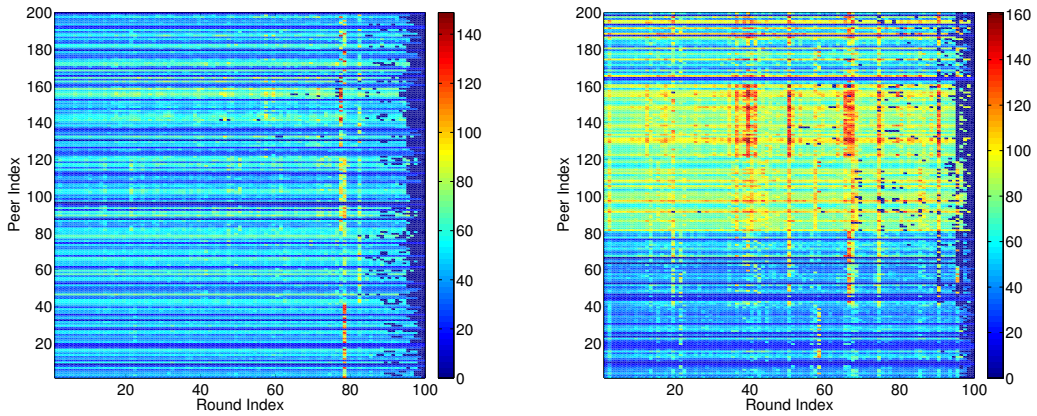
$$v_{ij} = \Xi - L_s(n_i) - d(n_i, n_j),$$

for an arbitrary $\Xi \in \mathbb{R}_{\geq 0}$ large enough to avoid having negative values in the simulation. After the basic performance indicators detailed in the previous section are calculated, the peercaster generates a new chunk and makes it available for download. Then, every peer performs peer selection according to Algorithm 3 (see Section 5.5.4), with each peer maintaining a list of its received bids. After bidding has finished, each peer sorts its list of received bids according to bid value, rejecting all but the top M_i bids where M_i is its upload capacity in chunks per round. After the winning bids are selected, any relevant measurement variables are updated, along with the chunk availability and delay maps of each peer.

The results of running the aforementioned simulation in the six delay scenarios of Section 5.5.5.1 yield the graphs presented in Figure 5.11. In Figure 5.11(a), we see that lag stretch was the smallest for the centred case of the King and clustered scenarios, and significantly longer for the rest. Overall, however, lag stretch was low: fewer than 20% of the peers have overlay delays more than 30% longer than a direct download from the peercaster, even in the worst case (the off-centre King delay distribution scenario). It is interesting to note that the protocol has a much higher propensity for the selection of longer delay paths in the off-centre King scenario when compared with the others, and this is because of the aggressive local value maximisation behaviour for each peer, which in this delay scenario yields *curved* end to end paths, rather than "straight" ones (see Figure 5.10(f)).

With respect to chunk lag (see Figure 5.11(b)), the protocol under-performs for the off-centre scenarios due to the large number of hops that the emergent distribution topologies have in those cases - again the off-centre King scenario is the worst performing, for the same reason as before. The chunk lag performance of the protocol was very similar for both the centred King and centred clustered delay distributions, and slightly worse for the centred uniform scenario. An interesting detail in Figure 5.11(b) is that the chunk lag curve for the off-centre uniform scenario has a large discontinuity at a chunk lag of 11 chunks, suggesting that a large proportion of the peers had this exact chunk lag. This can easily be explained using the *availability/stream delay* chunk map for this scenario.

The availability/stream delay chunk maps for both scenarios using a uniform peer distribution are shown in Figure 5.9. Each one of these maps has, over the horizontal axis, the indexes corresponding to all the rounds in a simulation up to a given time - in particular, the two chunk maps shown cover the first 100 rounds of their respective simulation. Over the vertical axis, each map has the indexes corresponding to all the peers participating in a given simulation. The colour that each coordinate has corresponds to the stream lag for that chunk, when the peer received it (the colourbar beside each plot shows the relation of lag with graph colour). We see that the protocol exhibits, in both scenarios, rounds where there was increased loss and delay (these are easily visible as vertical lines in the maps). These episodes, which occur with greater frequency in the uniform off-centre scenario, might be associated with cascading failure scenarios where a peer fails to secure a given chunk because its "usual" provider has consumed its upload bandwidth without uploading the chunk, which means that the entire distribution overlay depending on it will have to get the chunk from higher delay sources.



(a) Uniform centre delay distribution

(b) Uniform off-centre delay distribution

Figure 5.9: Availability and stream lag chunk maps

Figure 5.11(c) shows the leading edge continuity results for our six delay distribution scenarios. Again, the protocol under-performs for off-centred scenarios, which suggests that chunk diffusion is much more predictable on the cases where the peercaster is placed in the centre. An interesting detail we can see in the leading edge continuity curve for the clustered scenario with a centred peercaster is the presence of a distinct set of peers that have lower continuity, in the range $[0.84, 0.93]$. These might correspond to peers that rely on a larger set of uploaders, rather than a few high capacity ones, and consequently incur in higher variability.

With regard to peer currency balance (see Figure 5.11(d)), all topologies have approximately the same proportion of peers with positive balances and negative balances. From Figure 5.5.5.3, it is clear that not all peers are actively uploading chunks all the time - a distinction emerges by the execution of the algorithm between peers that become part of the "stream distribution infrastructure" and have higher upload utilisation, and purely consumer peers at the edges, that become incapable of further selling the chunks they buy and quickly enter into negative balances. Wealth distribution tends to be more unequal in the scenarios with centred peercasters than in the other ones, which is consistent with the previously discussed idea that the protocol is much better suited to diffuse chunks effectively in the scenarios with centred peercasters than in the others.

Our analysis of the number of bids that each peer rejects per round confirms the emergence of a small subset of peers that take most of the responsibility for chunk dissemination, and a much larger subset of peers that become passive consumers. The first important detail to observe in Figure 5.11(e) is that the CDF values at a lost bid rate of zero (for all delay distribution scenarios) were very similar, in the range $[.81, .93]$. This means that, in all cases, the proportion of peers that rejected bids was lower than 20%. Within the peers that did reject bids, however, the rates of bid rejection are extremely high, with all delay configurations having a small set of peers that rejected slightly more than one bid per node in the overlay per round. This is unsurprising, since all peers submit bids for all the chunks that they are missing, which means that the peercaster receives at least as many bids every round for its newly created chunk as there are peers in the overlay. This same phenomenon, to a lesser extent, happens for all peers in the distribution overlay.

The last metric we present is the synthetic performance metric (5.15). Since (5.15) will yield values close to 1 for those peers who are closest to the peercaster in lag, while having low variability on their chunk downloads, Figure 5.11(f) suggests, in agreement with our other measures, that the protocol was

much more successful in the three scenarios with a centred peercaster.

An overall conclusion from all the simulation runs performed is that the proposed protocol behaves much better in the scenarios where the peercaster is placed in the centre of the space, rather than those where it is placed on an edge. This could be construed as unsurprising, since so doing increases the the average delay from all peers to the peercaster, which might suggest a comparable decrease in overlay performance. However, even lag stretch, designed to take this into account by normalisation, shows quality degradation for off-centre scenarios.

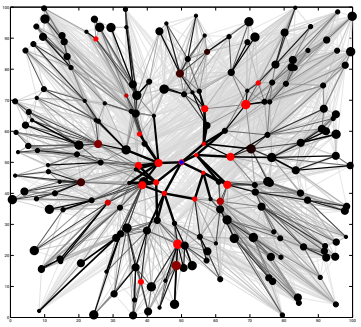
On the other hand, stream lag stretch is consistently good, indicating efficient distribution topologies. Overall, we believe that the simple example protocol presented demonstrates the power of the general model of Sections 5.1, 5.2 and 5.3.

5.6 Discussion of Potential Improvements

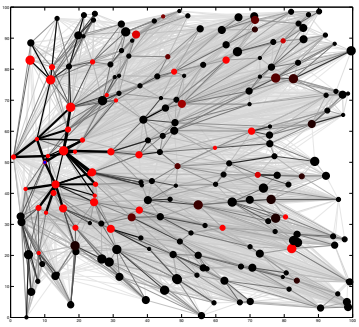
In this chapter we presented a swarming algorithm based in multi-item Vickrey auctions that balances the efficient construction of the QoS overlay with the requirements of an incentive mechanism.

With regard to possible improvements to model presented in this chapter, the first assumption that could be relaxed is that the peers themselves have a valuation function at all. One possible solution to this would be the peers constructing a valuation function by learning. Another option would be to change from a Vickrey auction model to a different one, where peers are unable to directly observe their own type, and must rely instead on (possibly noisy) signalling to infer it.

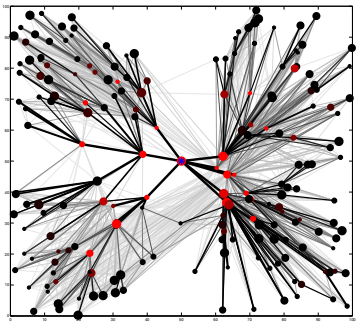
Finally, another possible improvement would be the substitution of the greedy utilisation-maximising peer selection algorithm of Section 5.3 with one that attempted a more complete optimisation procedure. Rather than making the locally optimal choice, some utility function could be optimised over the whole set of combinations of bid requests.



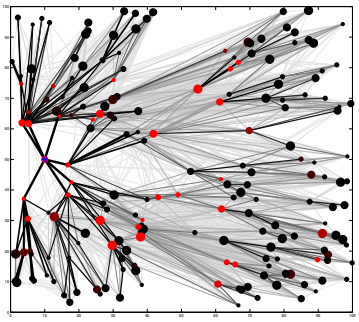
(a) Uniform delay distribution with centred peercaster



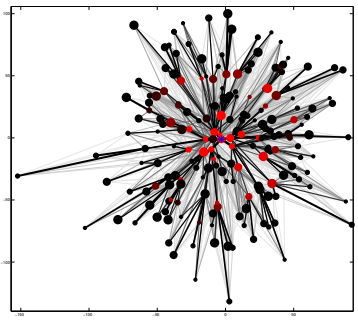
(b) Uniform delay distribution with off-centre peercaster



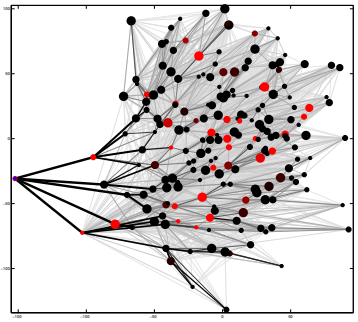
(c) Clustered delay distribution with centred peercaster



(d) Clustered delay distribution with off-centre peercaster

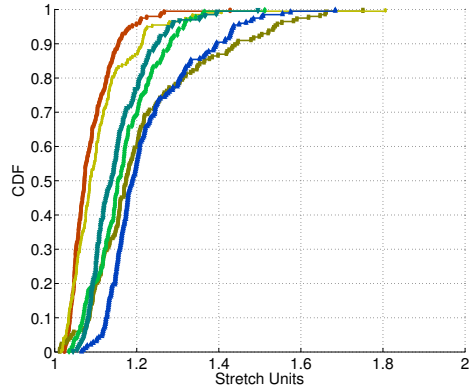


(e) King data subset delay distribution with centred peercaster

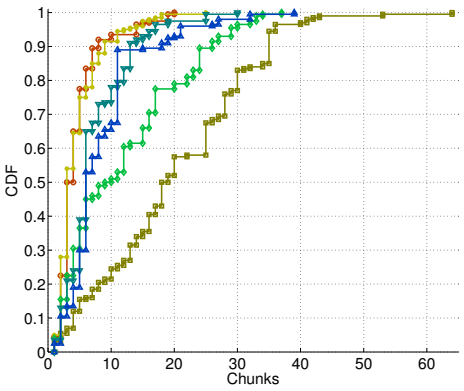


(f) King data subset delay distribution with off-centre peercaster

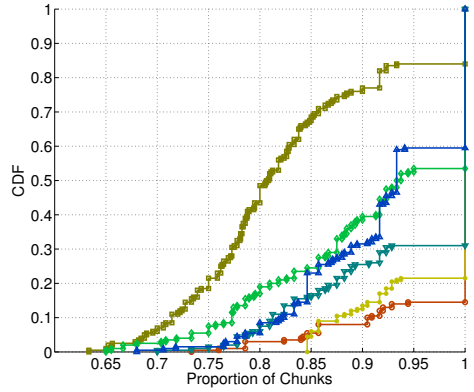
Figure 5.10: Peer delay space distributions and their resultant traffic patterns. In subfigures 5.10(a) and 5.10(b) peers are placed uniformly at random in delay space, in 5.10(c) and 5.10(d) they are placed so that four clear clusters can be found over the simulated delay space, and in 5.10(e) and 5.10(f) a subset of the King delay data set was extracted and embedded in a two dimensional coordinate system using the Euclidean metric and a square-error minimisation technique based on [90, 94]. The width and colour of links is indicative of the number of chunks exchanged between the peers at their endpoints after 100 simulation rounds. The size of the peer is indicative of upload bandwidth, and its colour of its upload bandwidth utilisation.



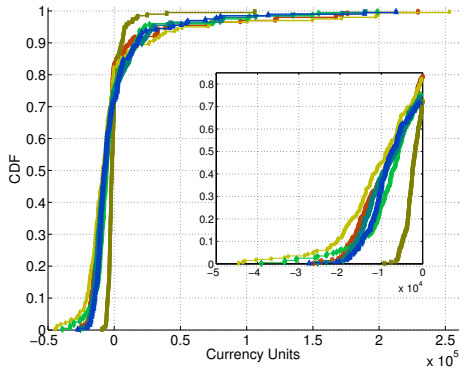
(a) Stream Lag Stretch



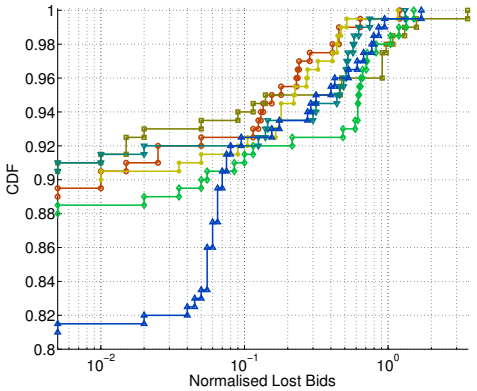
(b) Chunk Lag



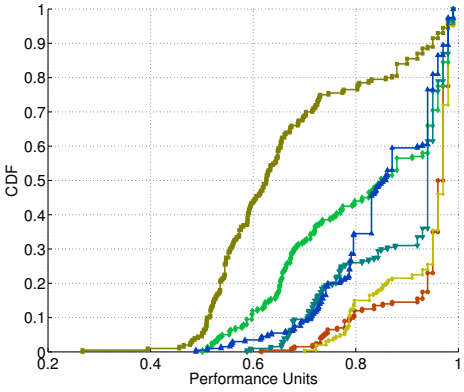
(c) Leading Edge Continuity



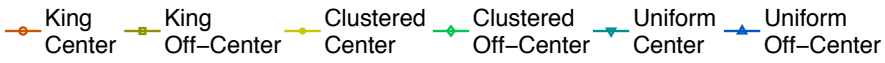
(d) Peer Currency Balance



(e) Lost Bid Rate



(f) Protocol Performance



(g) Legend

Figure 5.11: Simulation variables for selected delay configurations.

6

Hidden Action in Live Peer-to-Peer Streaming

Peer-to-peer networks providing QoS-enabled services are sensitive to *hidden action* situations, where the actions of a server peer are hidden from the peers who receive services from it. Thus, servers peers impose an *externality* on the clients: servers can choose to minimise their effort strategically, and client peers may be unable to distinguish between cases in which the server exerted insufficient effort and cases in which the server kept its advertised effort levels but the end-to-end conditions in the network were sufficiently adverse.

In this chapter, we propose a *principal-agent* model for hidden action that gives server peers sufficient incentives to meet their advertised effort levels, without client peers having to decide for each transaction whether the outcome was due to server behaviour or network conditions[1]. Essentially, our proposed algorithm forces server peers to accept part of the externality that would have been exclusively endured by the client peers. We achieve this by allowing client peers to draft optimal contracts that provide incentives for truthful revelation of QoS capabilities, and to have predictable transaction quality. We then exemplify the model for the case of a mesh-based, pull-oriented streaming system with low delay requirements. For this example, we treat each transfer of a delay-sensitive chunk as a *moral hazard* situation in which server peers can force negative externalities upon client peers. We show how to estimate the model parameters as a function of the prevailing network conditions, and how to enforce contract fulfilment through a reciprocative strategy on a repeated game. Finally, we show that a *probabilistic interaction tracker* can define the resulting Nash equilibrium to be a cooperative one by specifying the minimum discount parameter.

6.1 Exploiting Information Asymmetry

The design of QoS overlays over best-effort networks is still an open research problem [292, 308, 204]. A particular aspect of this is the problem of QoS as an observable indicator of peer behaviour: although the peer-experienced QoS will depend on the underlay network conditions, it will be also affected by the behaviour of the serving overlay peers. This creates a situation with *information asymmetry*: there is

[1] In this chapter we assume that peers have reliable identities, and therefore we do not address sybil or whitewashing attacks [105, 117].

information known to a peer (the true effort that a peer made in responding to requests), which is hidden to others (peers who receive services whose quality which might have been affected by the underlying network).

We propose an incentive mechanism that provides an incentive for server peers to deliver their advertised levels of effort, even if client peers are unable to determine unequivocally if the unsatisfactory outcome of a transaction is due to insufficient server effort or to network variability. We will focus on peer-to-peer overlays with strategic, selfish peers.

Although there is a large amount of work on the economic and game theoretical modelling of peer-to-peer systems (in particular as it relates to the *freeloader problem* [27, 116, 276, 114, 111]), it is frequently assumed that peers can determine the level of server effort by observing their own experienced QoS. Based on this, peers choose the service quality that they provide to other peers as a function of the service quality these other peers have, in turn, provided to the system.

Unfortunately, clients are very often unable to observe the actual server effort, since it is obscured by shifting network and peer conditions that affect transfer operations. The design for QoS contracts between peers connected through a best-effort network has been less widely studied.

The main difficulty in the analysis of peer-to-peer QoS contract design is that the effect of the behaviour of the server is *externalised* to the client. This means that if a server peer fails to deliver a high-enough contribution effort, and this leads to low QoS for the client, the server does not naturally bear the negative consequences of this action. Instead, these consequences are externalised to the peer who requested the service.

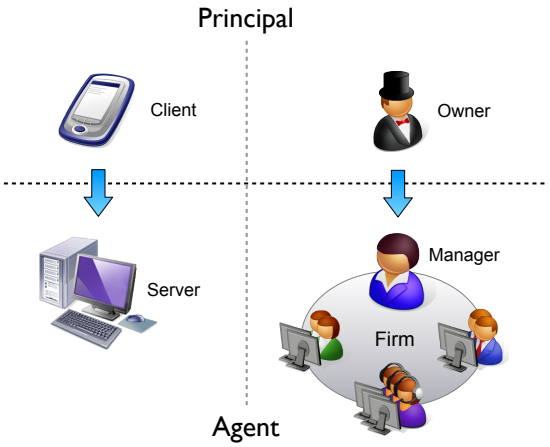


Figure 6.1: Applying the principal-agent model of economic theory to peer-to-peer overlays

Since the client is unable to measure directly the actions of the server, the fact that the server might have not delivered its advertised service quality can be only indirectly inferred. In this chapter, we propose a *principal-agent* model for hidden action that gives server peers sufficient incentives to meet their advertised effort levels, without them having to decide for each transaction whether the observable outcome (the actual QoS linked to the transaction) was due to server behaviour or network conditions. This is accomplished by allowing peers to draft contracts that provide incentives for truthful revelation of system contribution capabilities. This, in turn, helps strategic peers to obtain more predictable service levels.

We will apply the model presented in Section 3.4.1 to a client-server scenario with information asymmetry. We identify the server as the *agent*, the stakeholder whose task is to provide a service, but that can impose an externality on the other party through its behaviour. Consequently, we identify the client with the *principal*, which delegates the performance of a service to the agent. On this basis, we

propose an algorithm for the calculation of the probabilistically optimal payments on a QoS-dependent contract offering greater payment if the transaction is concluded with high QoS, and a lower payment otherwise. Although we propose a general expression for the calculation of these payments, every particular application will require its own probabilistic model of the relationship between the effort that the server puts into a transaction, the prevailing network conditions and the QoS experienced by the client. We present such a model for an specific example: a mesh-based, pull-oriented streaming system with low delay requirements.

6.2 Applying Hidden Action Models to QoS Overlays

Although the model briefly outlined in Section 6.1 is more general, in this thesis we focus on the modelling of quality-aware peer-to-peer overlays and hence restrict our attention to these.

In the case of peer-to-peer QoS overlays, the client plays the part of the principal and the server plays the part of the agent: the client trusts the server to provide its advertised effort, but the experienced QoS of the interaction will also depend on the variability of the network on which the service takes place. Thus, the server will have some degree of plausible deniability if its service is unsatisfactory, as degradation might be indicative of fluctuating network conditions and not lack of effort on its side.

In order to use the principal-agent model, we must assume the existence of a *market and currency system* $\mathcal{M}$. The role of $\mathcal{M}$ includes informing all peers of the effort guarantees advertised by other peers, communicating pricing and resource availability information, securely maintaining the amount of currency that peers possess, and executing currency transfers between peers. Functionally, $\mathcal{M}$ is the open market on which peers advertise their resources and formalise transactions. We assume that services are found, purchased and paid for using currency and procedures defined in $\mathcal{M}$.

There is a vast body of research regarding the distributed implementation of $\mathcal{M}$ using either structured or unstructured overlays [307, 309, 78, 302, 121, 313, 295, 328, 32], and our technique can be easily adapted to be used on any one of these systems. Instead of proposing our own implementation of $\mathcal{M}$, we propose an open model applicable to a wide range of possible $\mathcal{M}$ implementations. Of course, *PledgeRoute* (see Chapter 4) could be used as the payment mechanism for monetary transactions within $\mathcal{M}$.

Through $\mathcal{M}$, clients search for prospective servers according to the QoS characteristics that the servers themselves advertise. This takes the form of either a distributed hash table (DHT) query, or directed broadcast on an unstructured overlay. Of course, servers will only truthfully reveal their QoS capabilities if the benefit they can obtain if they do so is greater that the benefit they can obtain if they lie. We will explore this issue in Section 6.3.

A peer n_i with a server role will only provide a service to a client n_j if the utility that a transaction with n_j offers is at least as large as the average utility that n_i could obtain from a transaction with any other peer n_k. Then, if we define U_r as the average utility that a peer can expect as a result of a transaction, we have that the minimum utility that n_i will accept from an interaction with n_j is U_r (we call this the *rationality* condition).We assume that peers can query $\mathcal{M}$ to learn U_r, and that U_r is updated by $\mathcal{M}$ with every successful transaction.

Once n_i has accepted a request from n_j, it has to determine the treatment it will give to it: n_i might give different priority to different requests, or it might have other local processes competing for scarce CPU or upload bandwidth resources. Thus, n_i can respond to the request from n_j with various levels of *effort* $\phi \in \mathbb{R}_{\geq 0}$. The service quality $q \in \mathbb{R}_{\geq 0}$ that n_j experiences as an outcome of the transaction will be contingent on ϕ: higher values of q are positively correlated with higher values of ϕ.[2]

[2]We are deliberately vague in the definition of ϕ and q, as they are application-dependent. For some overlays ϕ and q might be defined in terms of delay, while for others they might be better defined in terms of jitter and average throughput. We provide a specific mapping of ϕ and q to time-sensitive chunk transfer in Section 6.3.

ϕ	Level of effort by the server
U_s	Utility function for the server
U_c	Utility function for the client
q	QoS enjoyed by the client
ψ	Payment given by the client to the server
p	Probability of the client getting high quality service as a function of the effort by the server.
U_r	Rationality utility for the server

Table 6.1: Principal-Agent model notation

However, we assume that the client is unable to observe ϕ - this is *private information* of the server. Additionally, we assume that there is uncertainty in QoS outcomes: due to varying delay and throughput conditions in the network, it is impossible for n_j to infer ϕ directly from the observed transaction QoS q. Therefore, an unobservable decision by n_i (its choice of ϕ) has an effect on the utility experienced by n_j, and we have a situation where the principal-agent model applies [228]. The client is unable to infer unambiguously the behaviour of the server from the observation of the transaction outcome, and is vulnerable to exploitation by it. This implies that the server needs to be given an incentive (in the form of an outcome-dependent payment $\psi \in \mathbb{R}_{\geq 0}$ measured in the currency units of $\mathcal{M}$) to give its best effort by assimilating some of the risk associated with the network conditions that might affect the transaction outcome. These variables are summarised in Table 6.1.

6.2.1 Mapping the Principal-Agent Model to QoS Overlays

The objective of the model is to obtain the payments that the client needs to offer the server as a function of the quality of service it receives, in order to ensure that the server puts maximal effort into maintaining its advertised effort levels. The client therefore calculates payments as a function of the server-advertised effort (and the estimated condition of the overlay network link between them) and creates a contract for the server. The server can either accept the contract as it stands, or reject it immediately.

If the server is able to maintain the effort commitments that it advertised in $\mathcal{M}$, it is considered to have devoted *high effort* to the transaction, and $\phi = \phi_+$. On the other hand, if the server is unable to maintain its advertised effort, it is considered to have devoted *low effort*, and $\phi = \phi_-$. Of course, low levels of effort increase the chance of the outcome having low QoS, and we seek to create an incentive mechanism against them[3].

A transaction is defined to have failed if the quality that the client experiences is lower than the minimum quality that the client is willing to tolerate, as set in the initial client-server contract. Thus, for any successful transaction, the outcome for the client can be either *high QoS* (and $q = q_+$) or *low QoS* (with $q = q_-$). We address transaction failure in Section 6.4.

We continue the derivation of the optimal contract payments by defining the utility function for the client (the principal) as

$$U_c(q, \psi) = q^\beta - \psi \quad \text{, where } \beta \in (0, 1),$$

and a corresponding utility function for the server (the agent)[4] as

$$U_s(\psi, \phi) = \psi^\alpha - \phi \quad \text{, where } \alpha \in (0, 1).$$

Theorem 6.1. *The solution for the principal-agent optimisation problem* (3.3) *with* $U_c(q, \psi) = q^\beta - \psi$

[3]The mapping between ϕ_+, ϕ_-, q_+ and q_- and observable parameters for an example peer-to-peer streaming application is analysed in Section 6.3.

[4]α and β are external parameters that can be used to fit these functions to experimental measurements. For the rest of this paper, we consider them external parameters that are given.

and $U_s(\psi, \phi) = \psi^\alpha - \phi$ *is given by*

$$\psi_- = \left(U_r + \frac{p_+\phi_- - p_-\phi_+}{p_+ - p_-} \right)^{\frac{1}{\alpha}} \tag{6.1}$$

$$\psi_+ = \left(U_r + \frac{(1-p_-)\phi_+ - (1-p_+)\phi_-}{p_+ - p_-} \right)^{\frac{1}{\alpha}}. \tag{6.2}$$

Proof. In this case, the optimisation problem (3.3) can be stated as

$$\begin{aligned} \text{Maximise:} \quad & p_+(q_+^\beta - \psi_+) + (1-p_+)(q_-^\beta - \psi_-) \\ \text{Subject to:} \quad & p_+\psi_+^\alpha + (1-p_+)\psi_-^\alpha - \phi_+ \geq U_r \\ \text{And:} \quad & p_+\psi_+^\alpha + (1-p_+)\psi_-^\alpha - \phi_+ \geq p_-\psi_+^\alpha + (1-p_-)\psi_-^\alpha - \phi_-. \end{aligned} \tag{6.3}$$

Following (3.4), we construct the Lagrangian, which in this case is

$$\begin{aligned} \mathcal{L}_H(\psi_+, \psi_-) = \quad & p_+(q_+^\beta - \psi_+) + (1-p_+)(q_-^\beta - \psi_-) \\ & + \lambda(p_+\psi_+^\alpha + (1-p_+)\psi_-^\alpha - \phi_+ - U_r) \\ & + \mu((p_+ - p_-)(\psi_+^\alpha - \psi_-^\alpha) - \phi_+ + \phi_-), \end{aligned}$$

and obtain the first-order conditions

$$\alpha\psi_+^{(\alpha-1)}(p_+\lambda + \mu(p_+ - p_-)) = p_+$$

$$\alpha\psi_-^{(\alpha-1)}((1-p_+)\lambda + \mu(p_+ - p_-)) = (1-p_+)$$

$$p_+\psi_+^\alpha + (1-p_+)\psi_-^\alpha = U_r + \phi_+ \tag{6.4}$$

$$(p_+ - p_-)(\psi_+^\alpha - \psi_-^\alpha) = \phi_+ - \phi_-. \tag{6.5}$$

By obtaining ψ_+^α from (6.4) and substituting on (6.5), we find that

$$U_r + \phi_+ - \psi_-^\alpha = \frac{p_+(\phi_+ - \phi_-)}{p_+ - p_-},$$

which implies that

$$\psi_- = \left(U_r + \frac{p_+\phi_- - p_-\phi_+}{p_+ - p_-} \right)^{\frac{1}{\alpha}}. \tag{6.6}$$

Finally, by substituting (6.6) on (6.5) we find that

$$\psi_+ = \left(U_r + \frac{(1-p_-)\phi_+ - (1-p_+)\phi_-}{p_+ - p_-} \right)^{\frac{1}{\alpha}}.$$

□

6.3 Delay-sensitive Chunk Transfer in Streaming Systems

In this section, we focus on modelling hidden action in a *mesh-based*, *pull-oriented* peer-to-peer media streaming system. In mesh-based systems, the media stream is divided into constant-sized *chunks* at its source, and each one of these is transported through the overlay in a more or less independent fashion. In pull-oriented systems, every chunk is negotiated in individual interactions where a request for a specific

chunk is issued. We call each of these interactions a *transaction*. A transaction consists of a peer (the *client*) issuing a request to another peer (the *server*) for a chunk, and the server responding with either the chunk in question or a message refusing the request. Of course, we require no asymmetrical restriction in roles: peers can be simultaneously clients and servers to other peers. Additionally, peers freely select to which peers to upload and from which peers to download, and each chunk is individually requested through a logically separate operation. Examples of these include DONet [327], PULSE [253] or Bullet [194].

In order to simplify the analysis, we assume that signalling and stream traffic is exchanged using TCP over already open connections[5].

We assume that the servers advertise, using $\mathcal{M}$, the effort levels that they are willing to commit to. Finally, we assume that peers need to schedule chunk downloads from other peers in a predictable manner in order to be able to maximise their QoS. Therefore, each client peer is assumed to solve an optimisation problem to decide which servers to ask for which chunks as a function of the effort parameters that they advertise, and its local codec play-out requirements. Our problem is then to design an incentives system that ensures that the server peers respond according to their advertised effort levels.

We model a transaction as shown in Figure 6.2 (the variables therein are explained in Table 6.2). The total delay that a client experiences on a transaction is:

$$D = t_{RTT} + \frac{S_r}{T_c} + t_P + \frac{N_s S_c}{T_s} \tag{6.7}$$

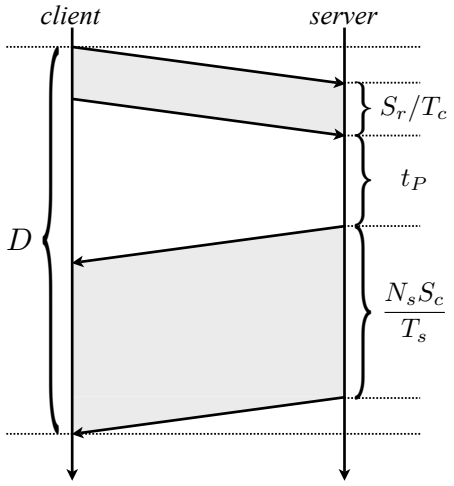


Figure 6.2: Components of the total transaction delay

Clearly, the variables in Table 6.2 fall into two kinds: those under the control of the server (t_P and N_s) and those that are a property of the protocol specification or the TCP connection between the client and the server (t_{RTT}, S_r, T_c and T_s). Thus, it is t_P and N_s that constitute the "advertised effort levels" that are disseminated through $\mathcal{M}$.

We model the effort put by the server on a given transaction by defining two deadlines: the first one, t_+, denotes the maximum transaction resolution time if the server actually delivers its advertised effort. The second deadline, t_-, is the absolute maximum delay that the client is willing to tolerate for the transaction. If a transaction was not rejected by the server immediately after its request, the elapsing

[5]We choose TCP only for analytical convenience. Extending the model to use other transport protocols, such as RTP [279] or TFRC [157], would require more explicit consideration of the channel loss probability, but is otherwise a simple matter. The performance of TCP for media streaming, however, has been analytically studied and found to be good if the achievable throughput is roughly twice the media bit-rate [310].

D	Chunk delivery delay
t_{RTT}	Layer 3 RTT
S_r	Request message size
T_c	Client to server throughput
S_c	Chunk Size
T_s	Server to client throughput
t_P	Request processing time at the server
N_s	Number of peers amongst which the server upload is shared

Table 6.2: Transaction delay components notation

S_p	Packet size at layer 3
p_L	Packet loss probability at layer 3
t_{RTO}	Retransmission timeout

Table 6.3: TCP Channel model notation

of t_- without the client having fully received the requested chunk will be interpreted by the client as deviation from a socially enforced rule, triggering penalties for the server (as detailed in Section 6.4).

To complete our characterisation of the behaviour of the client, we also define a time t_c during which the server has the opportunity to reject the transaction. If the client does not receive a rejection message before t_c elapses, it will silently assume that the server has accepted the request and it will deliver a chunk with the effort level that it advertised through $\mathcal{M}$.

Thus, we have defined a two-tier differentiated service scheme: if the observed delay D of the transaction is lower than or equal to t_+ (that has been calculated to be consistent with the server-advertised parameters), the outcome quality $q = q_+$ is considered to belong to the *high QoS* tier and $\psi = \psi_+$. On the other hand, if $t_+ < D < t_-$, the outcome quality $q = q_-$ is considered to belong to the *low QoS* tier and $\psi = \psi_-$. Finally, if $D > t_-$ the transaction is considered to have failed, and the server is penalised.

In order to apply the model in Section 6.2.1 (and in so doing obtain the optimum contract prices ψ_- and ψ_+), peers need to estimate p_+, p_-, ϕ_+ and ϕ_- as a function of the prevailing network conditions and the desired deadline for the transaction, t_+.[6]

To accomplish this, we require a model linking the transaction-level delay (the process that defines the QoS received by clients) with the packet-level delay (the varying network conditions). Thus, we seek to express the distribution the transaction delay in terms of the relevant parameters of the round trip packet delay and the model (6.7) (Figure 6.2). To do this, we note that the RTT delays can be modelled using a shifted gamma distribution [234, 54], so that the two shape parameters of the gamma distribution (θ and k), along with a shift parameter Δ can be fit from data on the prevailing characteristics of the RTT delay on the link. This means that we can express the RTT delay probability density function as

$$\mathsf{r}(x) = \frac{1}{\theta^k \Gamma(k)} (x - \Delta)^{k-1} \exp\left(-\frac{x - \Delta}{\theta}\right), \tag{6.8}$$

where $\Gamma(k)$ is the *Gamma function*,

$$\Gamma(k) = \int_0^\infty t^{k-1} e^{-t} dt,$$

and the shift parameter Δ, the scale parameter θ and the shape parameter k can then be obtained from RTT measurements using maximum likelihood estimation [74].

[6] Peers define t_- in a completely private fashion.

We require a model linking the strategic behaviour of the server and the expected behaviour of the network. If we assume that t_{RTT} is constant for the entirety of the transaction, that there are identical TCP stack configurations in the client and the server, and that TCP sharing is approximately fair amongst flows in the server, we can use the TCP model from [246] to obtain an expression for the transport layer throughput. If we use the nomenclature in Tables 6.2 and 6.3, we have that

$$T_c = T_s = \frac{S_p}{t_{RTT}\sqrt{\frac{2p_L}{3}} + t_{RTO}(3\sqrt{\frac{3p_L}{8}})p_L(1+32p_L^2)}. \tag{6.9}$$

Substituting (6.9) in (6.7) we have that

$$D = \xi_1 t_{RTT} + t_P + \xi_2, \tag{6.10}$$

where

$$\xi_1 = 1 + \sqrt{\frac{2p_L}{3}}\frac{S_r + S_c}{S_p} \quad \text{and} \tag{6.11}$$

$$\xi_2 = t_{RTO}(3\sqrt{\frac{3p_L}{8}})p_L(1+32p_L^2)\frac{S_r + N_s S_c}{S_p}. \tag{6.12}$$

By considering the relationship between the probability density (PDF) and the cumulative distribution (CDF) [268] of a random variable X with density $\mathsf{r}(x)$, we see that the the PDF of the random variable T obtained from X through a mapping h so that $T = h(X)$ is given by

$$\mathsf{d}(t) = \mathsf{r}(h^{-1}(t))\left|\frac{\partial(h^{-1}(t))}{\partial t}\right|. \tag{6.13}$$

Thus, if we consider the network RTT as a random variable X with a density given by (6.8), then the random variable T representing the total transaction delay is $T = h(X)$, with h being given by (6.10). Therefore, we have that

$$h^{-1}(t) = \frac{t - t_P - \xi_2}{\xi_1} \tag{6.14}$$

and thus, directly composing (6.14) with (6.8) and applying (6.13), we obtain

$$\mathsf{d}(t) = \frac{1}{\theta^k \Gamma(k)}\left(\frac{t - t_P - \xi_2}{\xi_1} - \Delta\right)^{k-1}\exp\left(-\frac{\frac{t-t_P-\xi_2}{\xi_1} - \Delta}{\theta}\right)\frac{1}{\xi_1},$$

which can be directly simplified to

$$\mathsf{d}(t) = \frac{1}{(\xi_1\theta)^k\Gamma(k)}(t - t_P - \xi_2 - \xi_1\Delta)^{k-1}\exp\left(-\frac{t - t_P - \xi_2 - \xi_1\Delta}{\xi_1\theta}\right). \tag{6.15}$$

For the cumulative distribution, $\mathsf{D}(t) = \mathbb{P}[T \leq t] = \int_0^t g(y)dy$, and we have that

$$\mathbb{P}[y < t] = \frac{1}{\Gamma(k)}\gamma\left(k, \frac{t - t_P - \xi_2 - \xi_1\Delta}{\theta\xi_1}\right),$$

where $\gamma(k, x)$ is the *lower incomplete gamma function*:

$$\gamma(k, x) = \int_0^x t^{k-1}e^{-t}dt.$$

In summary, by transforming the RTT density $\mathsf{r}(x)$ in (6.8) according to (6.7) and introducing a simple model for TCP throughput as parametrised by RTT [246] as shown in (6.9), we find that the transaction-level delay probability density $\mathsf{d}(t)$ is (6.15) where ξ_1 and ξ_2 are given by (6.11) and (6.12) respectively (see Tables 6.2 and 6.3 for nomenclature). We see from (6.15) that the transaction level delay probability density $\mathsf{d}(t)$ is also a gamma distribution with the same shape parameter k, but shifted to the right by $t_P + \xi_2 + \xi_1\Delta$ (compared to the shift parameter Δ for the RTT distribution $\mathsf{r}(x)$), and with scale parameter $\theta\xi_1$ (compared to just θ for $\mathsf{r}(x)$). Thus, it is to be expected that the transaction-level delay will have increased mean and variance, when compared to the simple RTT delay. This last observation stems from the fact that both $t_P + \xi_2 + \xi_1\Delta > \Delta$ and $\theta\xi_1 > \theta$, since by (6.11) we have that $\xi_1 > 1$, and both t_P and ξ_2 are positive and greater than zero.

For the rest of the parameters in the model, we define the mapping between the observable variables[7] and ϕ and q as given by

$$\phi = \kappa_1 \left(\frac{1}{N_s t_P}\right)^{\epsilon} \quad q = \kappa_2 \frac{1}{D^{\eta}}, \tag{6.16}$$

where $\kappa_1 \in \mathbb{R}_{\geq 0}, \kappa_2 \in \mathbb{R}_{\geq 0}, \eta \in (0,1), \epsilon \in (0,1)$. For ϕ_+ and q_+, the server is assumed to have operated according to its advertised parameters, and we have that

$$\phi_+ = \kappa_1 \left(\frac{1}{N_s^+ t_P^+}\right)^{\epsilon} \quad q_+ = \kappa_2 \frac{1}{D_+^{\eta}}.$$

To calculate t_+, the client queries $\mathcal{M}$ for the server-advertised t_P^+ and N_s^+ and defines a target p_+ that represents the tolerance that the client is willing to give to the server ($1 - p_+$ is the probability that the transaction will conclude after t_+ even though the server upholds its advertised t_P^+ and N_s^+, and therefore also measures the risk involved for the server). From (6.8), and referring to Figure 6.3, we see that t_+ is found by solving

$$\Gamma(k)p_+ = \gamma\left(k, \frac{t_+ - t_P^+ - \xi_2 - \xi_1\Delta}{\theta\xi_1}\right),$$

where $\Gamma(k)$ and $\gamma(k,x)$ are, again, the gamma and lower incomplete gamma functions.

We now model the two classes of service related to the effort levels ϕ_+ and ϕ_-. We refer the reader to Figure 6.3 for a graphical representation of the time intervals and probabilities related to the following discussion.

The transaction delay distribution f_+ associated with high-QoS can be obtained from (6.15) by setting $t_P = t_P^+$. In order to obtain f_-, we propose that if the server fails to uphold t_P^+ and N_s^+, it will choose an alternative effort level defined by t_P^- and N_s^- in which it attempts to put as little effort as possible, while trying to maintain high payment ψ_+. For the purpose of this chapter, we assume that the server reveals N_s truthfully in all circumstances, and thus $N_s^- = N_s^+$. Consequently, for ϕ_-, the server will decrease its effort by increasing its processing delay so that $t_P = t_P^- < t_P^+$, and it will calculate t_P^- so that the expected value of f_- is equal to t_+, therefore selecting the lowest effort that, in expectation, still leads to a high paying outcome. This means that the client chooses the t_P^- (see Figure 6.3) so that

$$\xi_1\Delta + \xi_2 + t_P^- + k\theta\xi_1 = t_+. \tag{6.17}$$

Clearly, the service level defined by (t_P^-, N_s^+) is significantly worse than that one implied by

[7] κ_1, κ_2, γ, ϵ and η are external parameters that can be used to fit these functions to experimental measurements. For the rest of this chapter, we consider them external parameters that are given.

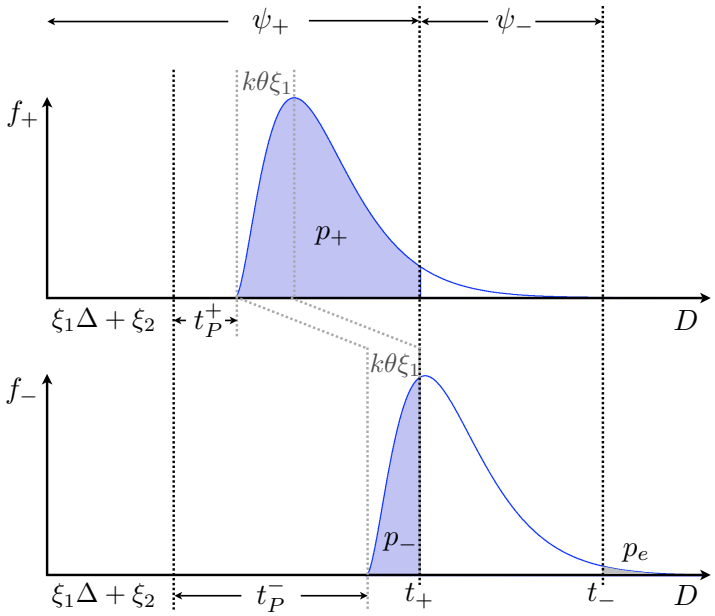


Figure 6.3: Determination of t_P^+ and t_P^-

(t_P^+, N_s^+), as normally the expected transaction resolution time if the server upholds t_P^+ is smaller than t_+. Once that t_P^- is defined, we calculate p_- by setting $t_P = t_P^-$ in (6.15) and finding the integral up to t_+, so that

$$p_- = \frac{1}{\Gamma(k)}\gamma\left(k, \frac{t_+ - t_P^- - \xi_2 - \xi_1\Delta}{\theta\xi_1}\right) = \frac{\gamma(k,k)}{\Gamma(k)}.$$

For ϕ_- and q_-, since the client assumes that the server has actually tried to deliver as low quality of service as it can get away with, we have that

$$\phi_- = \kappa_1\left(\frac{1}{N_s^+ t_P^-}\right)^{\epsilon} \quad q_- = \kappa_2\frac{1}{D_-^{\eta}}.$$

Clearly, it is in the best interest of the server to minimise its probability of being paid ψ_-. Thus, the server will only accept contracts for which it can deliver high values for p_+, which in turn gives it an incentive to reveal truthfully the effort level that it is willing and able to maintain.

Depending on the the value that the client assigns to t_-, there might be a possibility that the transaction could exceed t_-, with no malicious intent by the server. If this probability, p_e (see Figure 6.3) presents an undesirably high risk, the server can choose to reject the contract and refuse service. the same considerations apply if p_+ is too low.

In Figure 6.4 we see a graphical representation of expressions (6.1) and (6.2) in the context of the model presented in this section. Figure 6.4 shows the payments ψ_+ that a client would offer to a server as a function of its advertised effort, for an scenario with parameters as shown in Table 6.4. As an example, a server peer that advertises in $\mathcal{M}$ that it is able to commit to an effort level corresponding to $N_s^+ = 4$ and $t_P^+ = 100$ msec. can expect to be offered by a prospective client a contract where $\psi_+ \approx 10^{3.716} \approxeq 5196$ and $\psi_- \approx 10^{3.3} \approxeq 1995$ in $\mathcal{M}$ currency units. Thus, the contract would be drafted with these two payment values. Following the outcome of the transaction, the server will receive ψ_+ if the transaction is successful with high QoS, ψ_- if the transaction is successful with low QoS, and nothing if the transaction is unsuccessful.

It is clear that in both Figures 6.4 and 6.5, there are data points missing for some of the curves corresponding to low values of N_s. The reason for this is that in optimisation problem (6.3), there is

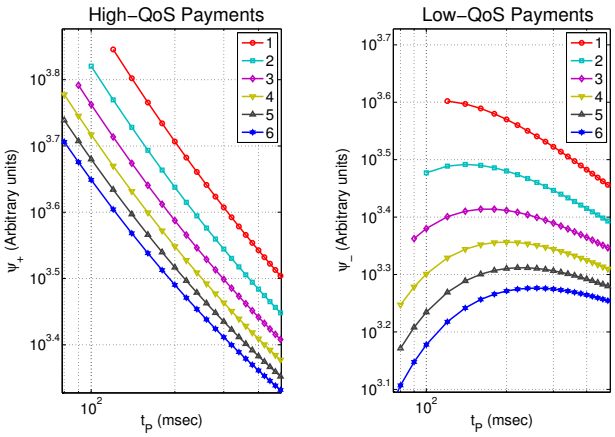


Figure 6.4: ψ_+ and ψ_- for the simulation run of Table 6.4 (the legend in both graphs shows different values of N_s, the number of clients that the server is serving simultaneously).

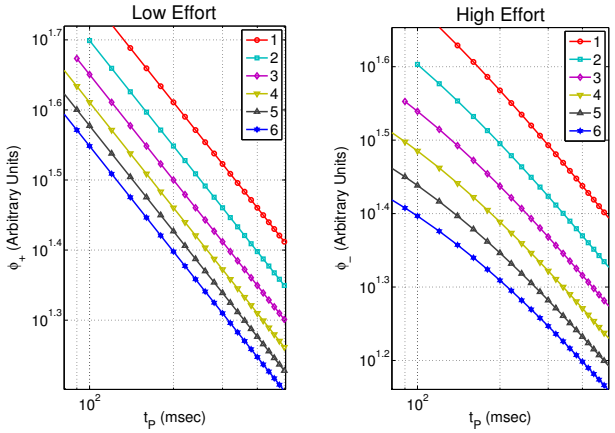


Figure 6.5: ϕ_+ and ϕ_- for the simulation run of Table 6.4 (the legend in both graphs shows different values of N_s, the number of clients that the server is serving simultaneously).

not an explicit requirement for the utility of the client to be positive. In fact, it is possible that the fees required to provide an incentive to the server will drive the client utility below zero. These cases are dropped, as is understood that the client will not have an incentive to request services from the server in the first place.

The heavily nonlinear behaviour of ψ_- near the origin is a product of the sensitivity of our definition for ϕ to very small decreases of t_P in the vicinity of zero, as it is clear from (6.16).

6.4 Contract Enforcement

The model in Section 6.2.1 assumes that the contract is enforceable, as it often the case in economics (usually through reliance on a legal framework). However, in the case of peer-to-peer overlay networks, such enforcement is difficult to provide because peers can lie about transaction outcomes, and this can be exploited to execute denial of service attacks against particular peers, greatly increasing the policing costs associated with each transaction. The application of well-known techniques of community enforcement [187] is complicated by the fact that peer interactions are unobservable beyond the relevant actors for the transaction, and therefore open to slander attacks. Instead, we propose a scheme based on the creation of *social capital* [86] by directly controlling the interaction graph of the peers. We model this peer-based enforcement as reciprocity-based trigger strategies on a repeated game.

Mean t_{RTT}	110 msec
k	3
θ	3
S_r	200 bytes
S_c	64 kbyte
S_p	1500 bytes
p_L	10^{-3}
t_{RTO}	100 msec
U_r	30
$\alpha = \eta = \epsilon$	.5
β	.8
τ	5
κ_1	1000
κ_2	1.5811×10^6

Table 6.4: Principal-Agent model simulation parameters

	Server Cooperate	**Server Defect**
Client Cooperate	(U_c^+, U_r)	$(-\tau, 0)$
Client Defect	$(U_c^d, -\phi_+)$	$(-\tau, 0)$

Table 6.5: Repeated game, normal form for moral hazard (Principal-Agent model)

We assume that peers follow the algorithm in Section 6.2.1 for the creation of contracts, which are susceptible to *moral hazard*[8]. We then proceed to model contract compliance as a repeated game where each stage game has the normal form shown in Table 6.5, where U_c^d, the *defection utility*, corresponds to

$$U_c^d = p_+ q_+^\beta + (1 - p_+) q_-^\beta,$$

and τ is a measure of time lost to the client waiting for a server response before timeout.

We model each transaction as a stage in a repeated game. We assume peers to have two strategies: adherence to the protocol, or selfish utility maximisation. Following the usual nomenclature for the *Prisoner's Dilemma*, we call adherence to the protocol *cooperation* and deviation from it (failure to uphold contractual obligations) as *defection*.

If $\mathcal{M}$ ensures that peers maintain a relatively stable relationship, so that the probability of peers interacting with each other remains under its control, it is possible to ensure that it is nonprofitable for the peers to fail to uphold their contracts.

To guide the Nash equilibrium of this repeated game, $\mathcal{M}$ acts as a *probabilistic interaction tracker* by explicitly modifying the information that it provides to the peers in order to ensure a minimum probability of repeated interaction δ (usually called the *discount parameter* in game theory). By forcing the δ between peers to the value where deviation from the protocol becomes unprofitable, we can ensure that only irrational peers will deviate. We explore two possible trigger strategies for contract enforcement: *Grim Trigger* and *Tit-for-Tat* [39].

6.4.1 *Grim Trigger* Enforcement

As usual, in this strategy a given peer n_i follows the protocol towards another peer n_j as long as n_j follows the protocol towards it. If n_j deviates, however, n_i retaliates by following a strategy of continuous defections from then onwards. This means that any protocol deviation receives the harshest punish-

[8] In situations with moral hazard, agents modify their behaviour toward risky activities depending on whether the consequences of these risks apply to themselves or to other agents.

ment possible, and there is no possibility for "forgiveness" and re-introduction of a possible cooperative regime. In this case, $\mathcal{M}$ must ensure that $\delta \geq \delta_G^{\min}$, where

$$\delta_G^{\min} = \frac{p_+\psi_+ + (1-p_+)\psi_-}{p_+q_+^\beta + (1-p_+)q_-^\beta + \tau}. \tag{6.18}$$

Proof. From the game normal form, it is clear that only the client has an incentive to defect. Without loss of generality, we consider the case where the client defects on the first round, and the server retaliates by defecting from then on. If we take into account the discount parameter δ defined by $\mathcal{M}$ as the probability that n_i and n_j will meet again to play the game, we have that for the expected utility for the client,

$$U^g = U_c^d - \frac{\delta}{1-\delta}\tau.$$

By considering the requirement that $U^g < U_c^+$, we have that

$$p_+q_+^\beta(1-\delta) + (1-p_+)q_-^\beta(1-\delta) - \delta\tau \leq p_+(q_+^\beta - \psi_+) + (1-p_+)(q_-^\beta - \psi_-),$$

and simplifying, we find that

$$\delta(p_+q_+^\beta + (1-p_+)q_-^\beta + \tau) \geq p_+\psi_+ + (1-p_+)\psi_-.$$

Thus, it is clear that

$$\delta \geq \frac{p_+\psi_+ + (1-p_+)\psi_-}{p_+q_+^\beta + (1-p_+)q_-^\beta + \tau}.$$

□

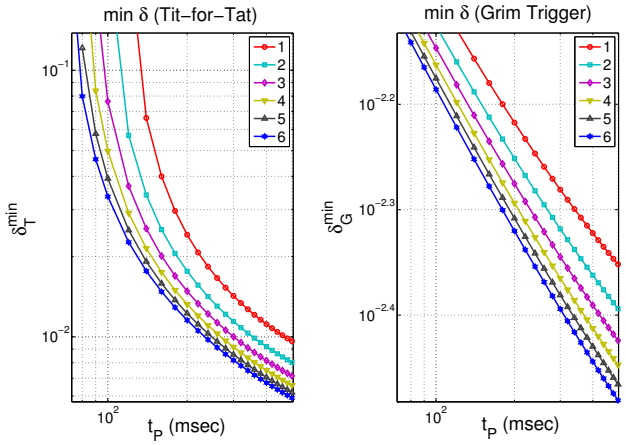


Figure 6.6: $\delta_T^{\min}$ and $\delta_G^{\min}$ for the simulation run of Table 6.4 (the legend in both graphs shows different values of N_s, the number of clients that the server is serving simultaneously).

6.4.2 *Tit-for-Tat* Enforcement

In this strategy a given peer n_i at a stage t applies the strategy that its partner n_j applied at a stage $t-1$. In this case, we have that $\mathcal{M}$ must ensure that $\delta \geq \delta_T^{\min}$, where

$$\delta_T^{\min} = \frac{p_+\psi_+ + (1-p_+)\psi_-}{p_+(q_+^\beta - \psi_+) + (1-p_+)(q_-^\beta - \psi_-) + \tau}. \tag{6.19}$$

Proof. In this case, the first defection from the client starts a series of alternating cooperation-defection episodes. For the expected utility for the client we have that

$$U^t = U_c^d - \delta\tau + \delta^2 U_c^d - \delta^3\tau + \ldots = (1 + \delta^2 + \delta^4 + \ldots)(U_c^d - \delta\tau).$$

Again, by considering the requirement that $U^t < U_c^+$, we have that

$$\frac{1}{1-\delta}U_c^+ \geq \frac{1}{1-\delta^2}(U^t - \delta\tau),$$

and simplifying, that

$$\delta \geq \frac{U^t - U_c^+}{U_c^+ + \tau}.$$

Finally, we see that

$$\delta \geq \frac{p_+\psi_+ + (1-p_+)\psi_-}{p_+(q_+^\beta - \psi_+) + (1-p_+)(q_-^\beta - \psi_-) + \tau}.$$

□

6.5 Discussion of Potential Improvements

In this Chapter we presented a contract technique to control hidden action situations in QoS overlays deployed over best-effort networks. An important issue that has not been directly addressed, however, is the distributed implementation of the probabilistic interaction tracker to which we refer in the Section 6.4.

By far the most complex issue surrounding the implementation of the probabilistic interaction tracker is the distributed, secure computation of (6.19) and (6.18). Although promising work has been done in the area of multiparty computation with strategic peers [112], the area is still open for contributions. Another related issue, the distributed implementation of a real time tracker, has been approached in the context of uniform node sampling [176], scalable anycast (see [235] and Section 2.4.1) or content-based routing [217]. We leave these improvements for future work.

7

Modelling Incentives in Managed Overlays

The previous three chapters of this thesis have been concerned with quality and incentives in peer-to-peer overlays in isolation from their underlying substrate. In Chapter 4, we defined the existence of two quality levels (*priority* and *best effort*), and we assumed that the peer in the role of server had total control over what the quality of service would be for the peer in the role of client. In Chapter 5, serving peers offer not only two levels of service, but as many levels as they require while maintaining their resource and incentives constraints. Finally, in Chapter 6 we address the issue of the service quality which each peer receives being a function not only of the quality allocated by the server peer, but also of the intervening network.

A common thread of these chapters, though, has been that the quality that the resources that the ISP substrate allocates to the overlay is taken as a given. However, by incorporating techniques such as P4P [320] and ALTO [252] it is possible to establish communication through the operator-overlay interface to more effectively drive resource allocation for services in the network edge.

7.1 ISP Multihoming and Managed Overlay Incentives

There has been growing interest in the exploration of communication and control interfaces between managed overlays and their underlying ISPs. The Application Layer Traffic Optimisation working group [252] focuses on providing an interface to ISP information that can be used to aid in peer selection for application layer overlays. In this chapter we assume a model like that shown in Figure 7.1, which consists of the following entities:

- **Edge Service Providers (ESP)**, managed overlays that provide QoS-based services on the network edge. Examples would be managed overlay multimedia operators (like PPLive) or content delivery networks such as Akamai or Limelight [22, 2, 24].

- **Network Providers (ISP)**, that provide network resources to end customers and ESPs. They can either own their own physical infrastructure and switching hardware, or purchase layer 2 services from an underlying set of telecommunications providers. We assume that they have complete

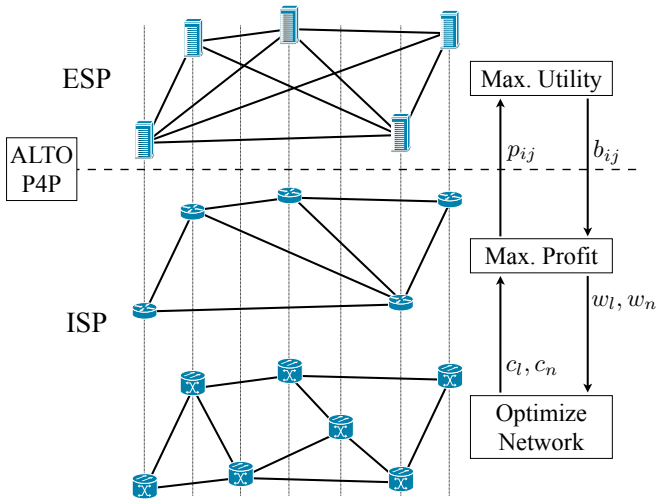


Figure 7.1: ALTO/P4P and an interface between overlay-based Edge Service Providers and ISPs.

control over their layer 3 infrastructure, possibly up to implementing their own QoS and routing algorithms.

The role of the ALTO/P4P interface in Figure 7.1 is to resolve the incentive tensions between the ESPs and the ISPs by providing each one with a way to express its preferences, which might be aligned with each other and can lead to mutually beneficial outcomes. We assume that this will be done in the simplest possible manner: the ISP will reveal its preferences through an *end-to-end flow price* p_{ji} between network endpoints j and i, and each overlay will take this into account to plan its own traffic matrix b_{ji} for all pairs of endpoints (j, i). Simplistically, the ISP will impose high prices on those routes for which it wants demand to fall, and it will decrease them for those for which it wants it to increase.

We are interested in the general case where the ESP is *multihomed*, and therefore can express its own preferences and respond to changing prices by reallocating its traffic between two given sites among a number of different ISPs[1]. Thus, we assume that every ESP has access to a set of ISPs, and that it is free to allocate its traffic freely between them. To make this preference expressivity by the ESP explicit, we assume that the ISP bills the ESPs for the source-destination flows between sites by volume, and according to the prices p_{ji} that were in effect when those flows took place.

We assume that each ESP will attempt to maximise the utility it gets from the connectivity provided by its underlying ISPs as a function of its own preferences, the network quality it experiences, and the end-to-end prices offered by them[2]. Each ISP, on the other hand, will attempt to maximise its profit as a function of the aggregated traffic matrix of all the ESPs it serves and its infrastructure link and router prices. Depending on the number of competing ISPs, the full range of monopoly/oligopoly/open-market situations can be realised on some ESP-ISP ecosystem.

The ISP might, in the simplest scenario, take the ESP traffic matrices as given, and calculate its end-to-end prices p_{ji} in an incremental fashion using an iterative technique such as supergradient ascent (as originally presented in [320]). However, it might also benefit from constructing a model of how a single ESP (or even the set of all ESPs) will respond to a given change in price. This more detailed model of the ESP price behaviour can be used as a component for ISP optimisation processes, in particular profit

[1]Of course, the practical implications of multihoming to policy routing, service survivability and the development of innovative business models go well beyond those we address. For brevity, we focus on the a very specific set of multihoming implications for the modelling of managed overlay incentives, as detailed in Section 7.2.

[2]Since we focus on managed overlays, we do not consider situations where each ESP member site or peer optimises its own utility separately

optimisation (as shown in Section 7.3.3).

Our contributions in this chapter are twofold. First, we propose a first-principles model for the price incentives of an ESP based on some simple assumptions regarding its preferences. Second, we provide a data-driven method based on least-squares regression to estimate the aggregate demand function of an ESP or a set of ESPs, and then show how can the ISP use this model to maximise its profit.

7.2 Preference Modelling of ESP Demand

The objective of this section is to develop, starting with elementary assumptions on the incentives of ESPs, a model for the price behaviour of ESPs that ISPs can use to predict the traffic matrix that an ESP will choose, given an input price per unit bandwidth for each origin-destination flow in the network. The definitions for the main elements of our model follow.

In our model, an *ESP* is formed by a set of *sites* N that reside at the network edge, usually in geographically and administratively distributed areas. These edge sites are assumed to wish to exchange traffic among them, forming a fully meshed overlay over one or more *ISP*s, each one formed by a set of routers $\mathscr{N}$ and a set of links $\mathscr{L}$ connected in any given topology. We will denote the set of all ISPs as $\mathcal{I}$, and the set of all ESPs as $\mathcal{E}$.

Further, in our model any ESP $m \in \mathcal{E}$ can decide to send traffic between any two sites (k, i) along any ISP $s \in \mathcal{I}$. This 3-tuple, consisting an origin site k, a destination site i and the ISP s used to communicate them will be called a *flow* $\xi = (s, k, i)$, and will be annotated with a *flow volume* b_{ski} which represents the amount of traffic that the flow carries, a *flow price* per unit bandwidth p_{ski} and a *flow quality* q_{ski}. The set of all possible flows between all sites will be denoted as L.

We start by assuming that each ESP will behave as a utility maximiser, taking the prices exposed by the ISP as input. We now formalise the ESP optimisation problem, assuming that each one of the sites of the ESP will have a utility U_i (see Table 7.1 for variable definitions) such that

$$U_i = \beta_i \left(\sum_{k \in \mathsf{N}, s \in \mathcal{I}} b_{ski}^{\alpha_i} q_{ski}^{\gamma_i} \right)^{\delta_i} - \sum_{k \in \mathsf{N}, s \in \mathcal{I}} p_{ski} b_{ski}. \tag{7.1}$$

The first term in (7.1) models the benefit that site i of an ESP obtains from the traffic it receives through all ISPs, and the second term the cost that it pays for it. Although the utility (7.1) is essentially arbitrary, it does capture many intuitions regarding the usefulness of an overlay-based service. We formalise these intuitions by analysing the following properties of our utility function.

- *Increasing utility with increasing traffic exchange between any two sites* ($\beta_i > 0$, $\alpha_i > 0$, $\delta_i > 0$). We assume that, if cost per unit capacity were not an issue, the protocol operations of the overlay operator might be significantly simplified by replicating all traffic at every site in the network. The cost savings that can be achieved by avoiding this and using intelligent stream processing algorithms has been one of the driving forces behind the development of locality-aware swarming protocols and distributed optimal load balancing techniques. This means that, in a cost-constrained scenario, being able to move more traffic between two overlay sites will result in an increased utility, albeit with diminishing returns measured by $1 > \alpha_i > 0$ and $1 > \delta_i > 0$.

- *Increasing utility with increasing quality on exchanges between any two sites* ($\gamma_i > 0$, $\delta_i > 0$). We assume that overlay links will be annotated with some notion of *quality* q_{ski}, so that transferring the same amount of traffic between two sites yields greater utility if the quality of the overlay link between them increases. This effect is particularly important for real-time content distribution overlays, such as media and gaming streams.

$U_i \in \mathbb{R}_{\geq 0}$	Utility that site i obtains from participation in the system
$b_{ski} \in \mathbb{R}_{\geq 0}$	Amount of traffic flow from site k to site i over ISP s
$q_{ski} \in \mathbb{R}_{\geq 0}$	Benefit that a unit bandwidth flow from site k to site i over ISP s provides to the ESP.
$p_{ski} \in \mathbb{R}_{\geq 0}$	Cost per unit bandwidth that the ESP must pay to transfer data from k to i over ISP s
$\mathcal{I}$	Set of all ISPs
$\mathcal{E}$	Set of all ESPs
$\mathscr{N}$	Set of all infrastructure routers in an ISP
$\mathscr{L}$	Set of all infrastructure links in an ISP
N	Set of all sites in the ESP
L	Set of all overlay links in an ISP (origin-destination pairs of sites)
$\alpha_i \in (0,1)$	Per-site diminishing traffic volume benefit for site i
$\gamma_i \in (0,1)$	Per-site diminishing traffic quality benefit for site i
$\beta_i \in (0,1)$	Total proportional traffic benefit for site i
$\delta_i \in (0,1)$	Total diminishing traffic volume benefit for site i

Table 7.1: ESP Utility model notation

- *ESPs pay an increasing cost with increasing traffic volumes exchanged between any two overlay sites* ($p_{ski} > 0$, $b_{ski} > 0$). We are assuming a simple pay-per-volume model where time is divided into discrete intervals of length T and each ISP s advertises source-destination prices per unit bandwidth p_{sji} between any two sites j and i that remain constant over the entirety of T. The situation where a ISP offers tiered services with differentiated costs can be easily addressed by decomposing the ISP into distinct ISPs, each one having a single quality class, and a single end-to-end price per unit volume for every overlay link.

- *Diminishing marginal utility on the amount of resources provided by a single site* ($\alpha_i < 1$). This models the fact that, if bandwidth costs are of importance, overlay sites can implement protocols where important data is exchanged with priority over less important data. Thus, any given site will have decreasing marginal usefulness for getting increasing amounts of traffic from a given overlay site.

- *Diminishing marginal utility on the quality that a given site is able to provide* ($\gamma_i < 1$). In many cases, such as voice or video streaming, once the quality of the received stream is high enough to decode the stream, no further improvement in playback quality will be achieved by increasing the transfer quality at the overlay network level.

- *Diminishing marginal utility on the number of sites that supply a site with resources* ($\delta_i < 1$). We assume that different overlay network sites might have access to different kinds of content or data of interest to other overlay sites, so that utility increases with the number of supplier sites that a given site has. However, it is improbable that all sites will yield equivalent usefulness to the requesting site, which will then contact them in decreasing order of usefulness. Consequently, this utility will also increase at a decreasing rate.

We assume that there will be competition between the different ESPs and between the different ESPs, so that neither an ISP monopoly or an ESP monopsony is assumed at the outset.

If we assume that the price that the ESP can charge its customers will be an increasing function of the utility of each one of their sites, then each ESP will function as a social welfare maximiser. Thus, its objective function will be the addition of all site utilities, and we can formulate the optimisation problem for the ESP as

$$\begin{aligned} \text{Maximise:} \quad U &= \sum_{i \in \mathsf{N}} U_i \\ &= \sum_{i \in \mathsf{N}} \left(\beta_i \left(\sum_{k \in \mathsf{N}, s \in \mathcal{I}} b_{ski}^{\alpha_i} q_{ski}^{\gamma_i} \right)^{\delta_i} - \sum_{k \in \mathsf{N}, s \in \mathcal{I}} p_{ski} b_{ski} \right). \end{aligned} \tag{7.2}$$

In our model, each ESP maximises the sum of the utilities of all its sites, where the utility of a site is given by (7.1). For this chapter, we will consider two cases in turn. In the first one the ESP is assumed to have unlimited wealth, an assumption that will be dropped for the second, more general case where we consider a budget constraint. Although only the latter case is important in practice, we present both because the solution for the second case relies on that of the first one.

7.2.1 Case 1: No Budget Constraints for the ESP

Here, we assume that the ESP has infinitely large currency reserves, and thus concentrate our attention on finding the utility-maximising traffic matrix that the ESP will submit to each ISP without cost restrictions. Since only non-restricted optimisation is required, we can apply first order conditions to (7.2) directly. By considering the optimality criterion $\nabla U = 0$, we have the following first order conditions, one for each end-to-end flow ji on each ISP s:

$$\frac{\partial U}{\partial b_{sji}} = \beta_i \alpha_i \delta_i \frac{q_{sji}^{\gamma_i} b_{sji}^{\alpha_i - 1}}{\left(\sum_{k \in \mathsf{N}, t \in \mathcal{I}} q_{tki}^{\gamma_i} b_{tki}^{\alpha_i} \right)^{1-\delta_i}} - p_{sji} = 0. \tag{7.3}$$

It is clear that the denominator in (7.3) does not depend on j, the origin of the flow, nor on its carrying ISP s - it is only a function of i, the site that is receiving the flow and hence assessing its utility. Thus, if we express (7.3) in terms of another, arbitrary flow terminating on the same site i but originating on a different origin site k and flowing through a different ISP y, and we take the ratio of the resulting expressions, the denominator and the destination-related multiplicative factors cancel out and we see that

$$\frac{q_{sji}^{\gamma_i} b_{sji}^{\alpha_i - 1}}{q_{yki}^{\gamma_i} b_{yki}^{\alpha_i - 1}} = \frac{p_{sji}}{p_{yki}}, \tag{7.4}$$

that can be rearranged to underscore the relationship between b_{sji} and b_{yki}, such that

$$\left(\frac{b_{sji}}{b_{yki}} \right)^{1-\alpha_i} = \frac{\frac{q_{sji}^{\gamma_i}}{p_{sji}}}{\frac{q_{yki}^{\gamma_i}}{p_{yki}}},$$

which means that, discounted by a diminishing returns exponent $1 - \alpha_i$, the ratio between the bandwidth allocated to two flows sji and yki terminating in the same site will be equal to the ratio between their **cost-benefits**, defined as the ratios between their qualities and their prices.

We now continue the derivation of the optimal b_{sji}. Going back to (7.4), we see that it can also be rearranged to give

$$\frac{q_{sji}^{\gamma_i} b_{sji}^{\alpha_i}}{q_{yki}^{\gamma_i} b_{yki}^{\alpha_i}} \left(\frac{q_{sji}^{\gamma_i}}{q_{yki}^{\gamma_i}} \right)^{\frac{1}{\alpha_i - 1}} = \left(\frac{p_{sji}}{p_{yki}} \right)^{\frac{\alpha_i}{\alpha_i - 1}}.$$

If we isolate the term involving b_{yki}, we find that

$$q_{yki}^{\gamma_i} b_{yki}^{\alpha_i} = \left(\frac{p_{yki}}{p_{sji}}\right)^{\frac{\alpha_i}{\alpha_i - 1}} \left(\frac{q_{sji}^{\gamma_i}}{q_{yki}^{\gamma_i}}\right)^{\frac{1}{\alpha_i - 1}} q_{sji}^{\gamma_i} b_{sji}^{\alpha_i},$$

and summing over k and y, we have that

$$\sum_{k \in \mathsf{N}, t \in \mathcal{I}} q_{yki}^{\gamma_i} b_{yki}^{\alpha_i} = q_{sji}^{\gamma_i} b_{sji}^{\alpha_i} \sum_{k \in \mathsf{N}, y \in \mathcal{I}} \left(\frac{p_{yki}}{p_{sji}}\right)^{\frac{\alpha_i}{\alpha_i - 1}} \left(\frac{q_{sji}^{\gamma_i}}{q_{yki}^{\gamma_i}}\right)^{\frac{1}{\alpha_i - 1}}.$$

Of course, the summation in the left is exactly the same one present in the denominator of (7.3), with only the indexes changed. Therefore, substituting in (7.3), we have that

$$q_{sji}^{\gamma_i} b_{sji}^{\alpha_i - 1} = \frac{p_{sji}}{\beta_i \alpha_i \delta_i} \left(\sum_{k \in \mathsf{N}, t \in \mathcal{I}} \left(\frac{p_{tki}}{p_{sji}}\right)^{\frac{\alpha_i}{\alpha_i - 1}} \left(\frac{q_{sji}^{\gamma_i}}{q_{tki}^{\gamma_i}}\right)^{\frac{1}{\alpha_i - 1}} \right)^{1 - \delta_i}. \tag{7.5}$$

By extracting p_{sji} and q_{sji} from the summation and simplifying we find that

$$b_{sji}^{\frac{1-\alpha_i\delta_i}{1-\delta_i}} = (\beta_i \alpha_i \delta_i)^{\frac{1}{1-\delta_i}} \frac{\left(\frac{q_{sji}^{\gamma_i}}{p_{sji}}\right)^{\frac{1-\alpha_i\delta_i}{(1-\alpha_i)(1-\delta_i)}}}{\sum_{k \in \mathsf{N}, t \in \mathcal{I}} p_{tki} \left(\frac{q_{tki}^{\gamma_i}}{p_{tki}}\right)^{\frac{1}{1-\alpha_i}}}, \tag{7.6}$$

which finally yields for b_{sji}, the optimal bandwidth allocation for a flow between sites j and i over ISP s, that

$$b_{sji} = (\beta_i \alpha_i \delta_i)^{\frac{1}{1-\alpha_i\delta_i}} \frac{\left(\frac{q_{sji}^{\gamma_i}}{p_{sji}}\right)^{\frac{1}{1-\alpha_i}}}{\left(\sum_{k \in \mathsf{N}, t \in \mathcal{I}} p_{tki} \left(\frac{q_{tki}^{\gamma_i}}{p_{tki}}\right)^{\frac{1}{1-\alpha_i}}\right)^{\frac{1-\delta_i}{1-\alpha_i\delta_i}}}. \tag{7.7}$$

This expression allows the ISP to estimate the optimal traffic matrix that an ESP might choose, given the prices exposed its underlying ISPs. Then, (7.7) is an estimated *best response* to any particular price vector p_{sji} by all the ISPs. As expected, the *cost-benefit* of flow sji plays a prominent role in (7.6) and (7.7).

We now continue our analysis of the ESP incentives by considering the total ESP utility for these optimal b_{sji}. From (7.3), we see that

$$\beta_i \alpha_i \delta_i \frac{q_{sji}^{\gamma_i} b_{sji}^{\alpha_i}}{\left(\sum_{k \in \mathsf{N}, t \in \mathcal{I}} q_{tki}^{\gamma_i} b_{tki}^{\alpha_i}\right)^{1-\delta_i}} = b_{sji} p_{sji},$$

and, by performing a summation over s and j, we find that

$$\sum_{j \in \mathsf{N}, s \in \mathcal{I}} b_{sji} p_{sji} = \beta_i \alpha_i \delta_i \left(\sum_{k \in \mathsf{N}, t \in \mathcal{I}} q_{tki}^{\gamma_i} b_{tki}^{\alpha_i} \right)^{\delta_i}, \tag{7.8}$$

where we recognise the summation on the left as the cost term from (7.2). If we define U^* as the optimal

ESP utility and substitute (7.8) in (7.2), we find that:

$$U^* = \sum_{i\in\mathsf{N}} \left(\beta_i \left(\sum_{k\in\mathsf{N},s\in\mathcal{I}} b_{ski}^{\alpha_i} q_{ski}^{\gamma_i} \right)^{\delta_i} - \beta_i\alpha_i\delta_i \left(\sum_{k\in\mathsf{N},t\in\mathcal{I}} q_{tki}^{\gamma_i} b_{tki}^{\alpha_i} \right)^{\delta_i} \right) \tag{7.9}$$
$$= \sum_{i\in\mathsf{N}} \beta_i(1-\alpha_i\delta_i) \left(\sum_{k\in\mathsf{N},s\in\mathcal{I}} b_{ski}^{\alpha_i} q_{ski}^{\gamma_i} \right)^{\delta_i}.$$

We now proceed to find the closed form for the inner summation in the simplified presentation of (7.9). By directly computing $b_{sji}^{\alpha_i}$ from (7.7), multiplying by $q_{sji}^{\gamma_i}$ and performing a summation over both s and j, we find that

$$\sum_{k\in\mathsf{N},t\in\mathcal{I}} q_{tki}^{\gamma_i} b_{tki}^{\alpha_i} = (\beta_i\alpha_i\delta_i)^{\frac{\alpha_i}{1-\alpha_i\delta_i}} \frac{\sum_{k\in\mathsf{N},t\in\mathcal{I}} q_{tki}^{\gamma_i} \left(\frac{q_{tki}^{\gamma_i}}{p_{tki}}\right)^{\frac{\alpha_i}{1-\alpha_i}}}{\left(\sum_{k\in\mathsf{N},t\in\mathcal{I}} p_{tki} \left(\frac{q_{tki}^{\gamma_i}}{p_{tki}}\right)^{\frac{1}{1-\alpha_i}}\right)^{\frac{\alpha_i(1-\delta_i)}{1-\alpha_i\delta_i}}},$$

which simplifies to

$$\sum_{k\in\mathsf{N},t\in\mathcal{I}} q_{tki}^{\gamma_i} b_{tki}^{\alpha_i} = (\beta_i\alpha_i\delta_i)^{\frac{\alpha_i}{1-\alpha_i\delta_i}} \left(\sum_{k\in\mathsf{N},t\in\mathcal{I}} p_{tki} \left(\frac{q_{tki}^{\gamma_i}}{p_{tki}}\right)^{\frac{1}{1-\alpha_i}} \right)^{\frac{1-\alpha_i}{1-\alpha_i\delta_i}}.$$

Substituting back in (7.9), we obtain for the optimal utility that

$$U^* = \sum_{i\in\mathsf{N}} \beta_i(1-\alpha_i\delta_i)(\beta_i\alpha_i\delta_i)^{\frac{\alpha_i\delta_i}{1-\alpha_i\delta_i}} \left(\sum_{k\in\mathsf{N},t\in\mathcal{I}} p_{tki} \left(\frac{q_{tki}^{\gamma_i}}{p_{tki}}\right)^{\frac{1}{1-\alpha_i}} \right)^{\frac{(1-\alpha_i)\delta_i}{1-\alpha_i\delta_i}}$$
$$= \sum_{i\in\mathsf{N}} \left({}^2\left[(1-\alpha_1\delta_i)(\alpha_i\delta_i)\right] \beta_i \left(\sum_{k\in\mathsf{N},t\in\mathcal{I}} p_{tki} \left(\frac{q_{tki}^{\gamma_i}}{p_{tki}}\right)^{\frac{1}{1-\alpha_i}} \right)^{(1-\alpha_i)\delta_i} \right)^{\frac{1}{1-\alpha_i\delta_i}},$$

where ${}^2x$ is the *tetration* function of order 2:

$${}^k x = \underbrace{x^{x^{\cdots^x}}}_{k \text{ times}}.$$

7.2.2 Case 2: Binding Budget Constraint for the ESP

We now address the case where the ESP has a budget restriction. In this case, the optimal solution to the problem is no longer a simple superposition of optima for each ISP. We propose an improved model which, as we shall see, is a simple extension from the model presented in Section 7.2.1. The optimisation problem central to this new model can be stated as

$$\text{Maximise: } U = \sum_{i\in\mathsf{N}} U_i = \sum_{i\in\mathsf{N}} \left(\beta_i \left(\sum_{k\in\mathsf{N},s\in\mathcal{I}} b_{ski}^{\alpha_i} q_{ski}^{\gamma_i} \right)^{\delta_i} - \sum_{k\in\mathsf{N},s\in\mathcal{I}} p_{ski} b_{ski} \right)$$
$$\text{Subject to: } \sum_{i\in\mathsf{N},k\in\mathsf{N},t\in\mathcal{I}} b_{ski} p_{ski} \leq \mathcal{B}, \tag{7.10}$$

where (7.10) is our budget constraint and $\mathcal{B}$ is the maximum amount of money that the ESP is willing to pay to the set of its underlying ISPs. The Lagrangian for this case,

$$\mathcal{L}_E = U + \lambda \left(\mathcal{B} - \sum_{i \in \mathsf{N}, k \in \mathsf{N}, s \in \mathcal{I}} b_{ski} p_{ski} \right),$$

leads to the following first order optimality conditions:

$$\frac{\partial \mathcal{L}}{\partial b_{sji}} = \beta_i \alpha_i \delta_i \frac{q_{sji}^{\gamma_i} b_{sji}^{\alpha_i - 1}}{\left(\sum_{k \in \mathsf{N}, t \in \mathcal{I}} q_{tki}^{\gamma_i} b_{tki}^{\alpha_i} \right)^{1-\delta_i}} - (1+\lambda) p_{sji} = 0. \tag{7.11}$$

The derivation proceeds as in the previous case, so that the equivalent expression to (7.5) is

$$q_{sji}^{\gamma_i} b_{sji}^{\alpha_i - 1} = \frac{(1+\lambda) p_{sji}}{\beta_i \alpha_i \delta_i} \left(\sum_{k \in \mathsf{N}, t \in \mathcal{I}} \left(\frac{p_{tki}}{p_{sji}} \right)^{\frac{\alpha_i}{\alpha_i - 1}} \left(\frac{q_{sji}^{\gamma_i}}{q_{tki}^{\gamma_i}} \right)^{\frac{1}{\alpha_i - 1}} \right)^{1-\delta_i}.$$

This expression can be re-arranged and simplified so that we obtain an expression for the b_{sji} in terms of the Lagrange multiplier λ, and as a result we have that

$$b_{sji} = (\beta_i \alpha_i \delta_i)^{\frac{1}{1-\alpha_i \delta_i}} \left(\frac{1}{1+\lambda} \right)^{\frac{1}{1-\alpha_i}} \frac{\left(\frac{q_{sji}^{\gamma_i}}{p_{sji}} \right)^{\frac{1}{1-\alpha_i}}}{\left(\sum_{k \in \mathsf{N}, t \in \mathcal{I}} p_{tki} \left(\frac{q_{tki}^{\gamma_i}}{p_{tki}} \right)^{\frac{1}{1-\alpha_i}} \right)^{\frac{1-\delta_i}{1-\alpha_i \delta_i}}}. \tag{7.12}$$

As expected, (7.12) reduces to (7.7) if $\lambda = 0$. Furthermore, if we define $\hat{b}_{sji}$ as the flow volume that *would have been allocated* between sites i and j in ISP s if the budget constraint had not been binding (as given by (7.7)), we have that (7.12) reduces to

$$b_{sji} = \left(\frac{1}{1+\lambda} \right)^{\frac{1}{1-\alpha_i}} \hat{b}_{sji}. \tag{7.13}$$

If we define $\mathcal{B}_i$ as the total cost that the ESP will have to pay to all ISPs for all the traffic terminating at site i, and $\hat{\mathcal{B}}_i$ as the total cost for all traffic terminating at i *had the budget condition not been binding*, we have that

$$\begin{aligned} \mathcal{B}_i &= \sum_{k \in \mathsf{N}, s \in \mathcal{I}} p_{ski} b_{ski} \\ &= \left(\frac{1}{1+\lambda} \right)^{\frac{1}{1-\alpha_i}} \sum_{k \in \mathsf{N}, s \in \mathcal{I}} p_{ski} \hat{b}_{ski} \\ &= \left(\frac{1}{1+\lambda} \right)^{\frac{1}{1-\alpha_i}} \hat{\mathcal{B}}_i. \end{aligned} \tag{7.14}$$

Equipped with (7.14), we now turn to the calculation of λ. In the case of a binding (7.10), we have that $\mathcal{B} = \sum_{i \in \mathsf{N}} \mathcal{B}_i$, and the multiplier λ can be obtained by solving[3]:

$$\sum_{i \in \mathsf{N}} \left(\frac{1}{1+\lambda} \right)^{\frac{1}{1-\alpha_i}} \hat{\mathcal{B}}_i - \mathcal{B} = 0. \tag{7.15}$$

[3] In our simulations, this equation is solved using standard trust-region methods [87].

From the previous exposition, the most convenient procedure that can be used to obtain the traffic matrix of a budget-constrained ESP is then as follows. First, the $\hat{b}_{ski}$ are calculated using (7.7) and aggregated into $\hat{\mathcal{B}}_i$ using (7.14). If $\mathcal{B} \geq \sum_{i \in \mathsf{N}} \hat{\mathcal{B}}_i$, the budget condition is not binding and $b_{ski} = \hat{b}_{ski}$. If, on the other hand, $\mathcal{B} < \sum_{i \in \mathsf{N}} \hat{\mathcal{B}}_i$, the budget constraint binds and we formulate (7.15) to find λ, which can then be used in (7.13) to find b_{ski} from λ and the previously found $\hat{b}_{ski}$.

It is easy to see, from (7.15) that $\mathcal{B} < \sum_{i \in \mathsf{N}} \hat{\mathcal{B}}_i$ implies that $\lambda > 0$. First, we note that since $0 < \alpha_i < 1$, the exponent for the expression containing λ will be, for all terms in 7.15, positive and larger than 1. Because of this, and since we have that all $\hat{\mathcal{B}}_i > 0$, in the case of a binding condition we require $\frac{1}{1+\lambda} < 1$, which in turn implies that $\lambda > 0$.

7.2.3 Simulation Experiments

To provide a more intuitive perspective on the properties on the model in the previous section, in Figure 7.2 we show the result of applying it to the following situation.

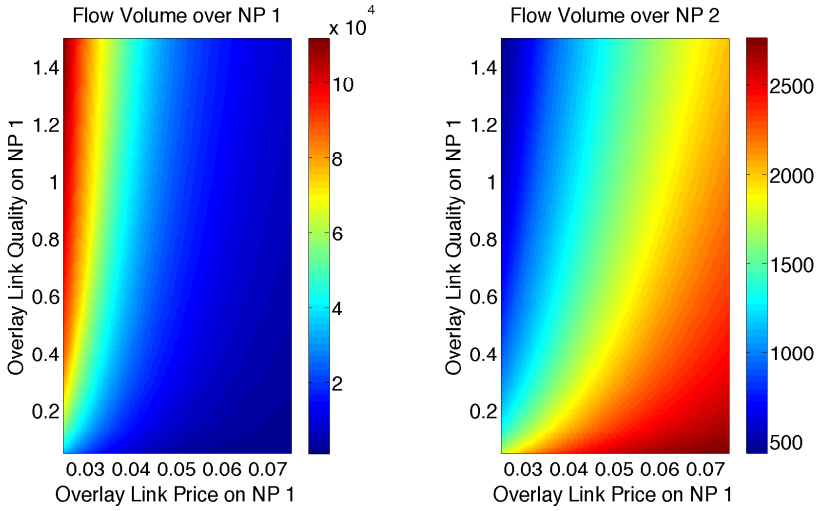
An ESP with 3 sites i, j and k has a choice of 2 ISPs to send traffic between them (the specific model parameters used are shown in Appendix B.1). We shall call these two ISPs **NP 1** and **NP 2**. We shall now vary the quality and cost of overlay link (i, j), from i to j, and see how the model adapts to this change.

The effects that changing the quality and price of (i, j) can be found in Figure 7.2(a). The graph in the left of Figure 7.2(a) shows the bandwidth allocated to (i, j) over **NP 1**, while the one on the right shows the bandwidth allocated to (i, j), but over the competing ISP **NP 2**. As the price of (i, j) increases in **NP 1**, the flow volume demanded by the ESP decreases very quickly initially, and then at a decreasing rate. As the cost-benefit for this link becomes smaller, the ESP re-allocates its traffic between the other flows to maximise its utility. In particular, the traffic along (i, j) but over **NP 2** increases, as its comparative cost-benefit improves. We see that something similar happens for quality: as the quality of the overlay link between the i and j decreases in **NP 1**, less traffic is sent through **NP 1** and more traffic is sent through **NP 2**. This implies that, qualitatively, price-quality combinations where (i, j) traffic over **NP 1** are higher imply that traffic between the same two sites but over **NP 2** are lower, and the ESP is effectively substituting decreased (i, j) flow volume over **NP 1** with increased overlay flow volume over all its other routes over **NP 1** and **NP 2**.

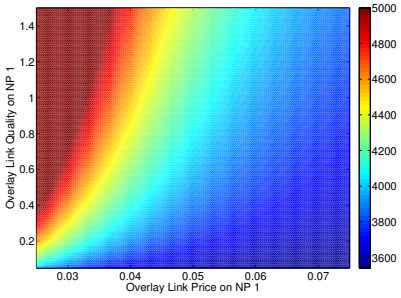
Figure 7.2(b) shows the total cost for that the ESP has to pay for all its traffic. First, we see that the budget cap of 5000 units comes into effect in the situation where link price is low and quality is high. This means that, if this is an accurate model of the behaviour of an ESP, the open multihoming model that we presented in Section 7.1 gives an incentive for ISPs to deliver better quality at a lower cost, since that in turn provides an incentive to users to spend more money. This is clearly visible in Figure 7.2(c), where the provision of high quality overlay links at low cost gives the ESP the highest utility.

Since we did not introduce any explicit traffic volume constraints in the model, we see that the reduction in traffic induced by increasing cost in the overlay link over **NP 1** is not balanced by an equivalent increase of traffic over **NP 2**. This is because the ESP is substituting, proportionally to cost-benefit and with diminishing returns, with traffic to all other destinations. Thus, having a high-quality, low-cost link is always preferable to the ESP than not having it, and the same general shape seen in the left of Figure 7.2(a) is seen on Figure 7.2(c), but with a more less defined cutoff. The ability of the ESP to redistribute its traffic between sites allows it some degree of resilience over price changes on a single link.

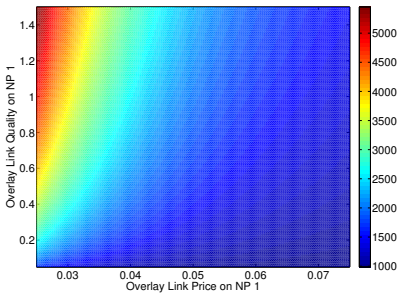
As a final comment on the model simulations, we note that all graphs seem to have a much stronger dependence on price than on quality, suggesting that for the parameter values chosen, ESPs will be more sensitive to price than to quality. Validating these results will experimental data will be the focus of future work.



(a) Link Bandwidth Allocations



(b) Total ESP Cost



(c) Total ESP Utility

Figure 7.2: Simulating quality and price changes for the ESP model

7.3 Experimental Fitting for the ESP Demand

The first-principles model of Section 7.2 has the advantage of giving some insight on the tradeoffs made by ESPs, but has the quite significant disadvantage of depending on the estimation of the utility parameters α_i, β_i, γ_i, δ_i and q_{ski} which include not only private information of each ESP, but also quality measurements on competing ISPs that might not be available. Therefore, a purely experimental curve-fitting approach might be more fruitful in practice. We will address the problem of finding the aggregate demand function for all ESPs of a given ISP, but the technique presented can be easily applied to estimate the individual demand functions of each ESP.

The procedure begins with the ISP aggregating the traffic matrices requested by each ESP and producing the total traffic matrix associated with all ESPs. We will find it useful to index end-to-end flows using a single index k instead of site duples (j, i). Thus, instead of dealing with the traffic matrix b_{sji}, we restrict to a single ISP s and deal with a endpoint-to-endpoint indexed vector $b_k \in \mathsf{L}$, where the *flow space* L is the set of all possible overlay links between sites, and k indexes all origin-destination pairs (i, j). If we aggregate the different traffic demand vectors provided by the ESPs, the ISP can obtain the total endpoint-to-endpoint traffic matrix B_k. Thus, if we define b_k^m as the load on the ISP along route k imposed by ESP m, we have that

$$B_k = \sum_{m \in \mathcal{E}} b_k^m.$$

To estimate an aggregate demand function for all its ESPs, the ISP will gather data relating the end-to-end price per unit flow that it offers to the ESPs, and the aggregate ESP traffic matrices that they

demand at those prices. Then, it will fit this data to a general functional form that is simple, yet flexible enough to capture the preference structure of the ESP aggregate. We propose to use the well-known Cobb-Douglas [83] function for this demand model, such that

$$B_k = A_0^k \prod_{\xi \in \mathsf{L}} p_\xi^{\eta_\xi^k}. \tag{7.16}$$

This model, if $\eta_0^k = \log(A_0^k)$, can then be expressed as

$$\log B_k = \eta_0^k + \sum_{\xi \in \mathsf{L}} \eta_\xi^k \log p_\xi. \tag{7.17}$$

The rationale behind this model is that we can explicitly model both the *price elasticity of demand* η_k^k and the *cross elasticity of demand* η_ξ^k for each end-to-end flow as a function of the prices of all other flows. This can be readily seen from the definition of the price cross elasticity of demand [144], which is

$$\frac{\partial \log B_k}{\partial \log p_\xi} = \eta_\xi^k.$$

7.3.1 Using Elasticity to Model ESP Demand

In microeconomics, the cross elasticity of demand measures the responsiveness of the quantity demanded of a good or service to changes in the price of *another* good or service. Therefore, the cross elasticity of demand can be used to define substitution relations between products or services: if increases in the price for a product decrease the quantity demanded of another product, the products are *complements* of each other, and will tend to be used together. Conversely, if increases in the price for a product increase the quantity demanded of another product, they are *substitutes* of each other, and will tend to be used instead of each other. If the price and quantity of two products are uncorrelated, their cross elasticity of demand will be zero and they are *independent* from each other (see Table 7.2 for a detailed taxonomy of the values that the cross elasticity of demand can take, and informal descriptions of their implications).

Thus, by explicitly modelling elasticities, we can model substitution effects between flows directly: when the cost for a particular set of end-to-end flows increases, the ESPs might compensate by increasing their consumption of the "cheap" routes and pulling back on consumption for the expensive ones. This means that the model will be detailed enough to model situations where changes in price provide an incentive to the ESP aggregate to substitute consumption of a flow with consumption of a different flow.

The cross elasticity of demand generalises the well-known price elasticity of demand, that measures the responsiveness that the quantity demanded of a good or service has to changes in its own price. Since the price elasticity of demand η_k^k corresponds simply to the cross elasticity of demand between a flow and itself, we require no additional framework to account for it. However, since demand is usually decreasing with price, the price elasticity of demand is usually negative and only the first 5 rows of Table 7.2 apply.

In order to estimate the elasticities η_ξ^k, we rely on the P4P-ALTO interaction between the ESPs and the ISP. This basically involves the ISP supplying a price vector, the ESPs evaluating their demand function, and returning the bandwidth they are willing to consume at that price vector. By aggregating and storing these values and performing a simple linear least squares regression, the ISP can estimate the aggregate demand function.

Range	Category	Responsiveness	Implications
$\eta_\xi^k \to -\infty$	Complement	Perfectly Elastic	The demand for flow k becomes zero with an infinitesimal price increase for flow ξ.
$-\infty < \eta_\xi^k < -1$	Complement	Elastic	The reduction in the demand for flow k given a price increase for flow ξ is large.
$\eta_\xi^k = -1$	Complement	Unitary Elastic	The reduction in the demand for flow k given a price increase for flow ξ is comparable.
$-1 < \eta_\xi^k < 0$	Complement	Inelastic	The reduction in the demand for flow k given a price increase for flow ξ is small.
$\eta_\xi^k = 0$	Independent	Perfectly Inelastic	The reduction/increase in the demand for flow k given a price increase for flow ξ is zero.
$0 < \eta_\xi^k < 1$	Substitute	Inelastic	The increase in the demand for flow k given a price increase for flow ξ is small.
$\eta_\xi^k = 1$	Substitute	Unitary Elastic	The increase in the demand for flow k given a price increase for flow ξ is comparable.
$1 < \eta_\xi^k < \infty$	Substitute	Elastic	The increase in the demand for flow k given a price increase for flow ξ is large.
$\eta_\xi^k \to \infty$	Substitute	Perfectly Elastic	The increase in the demand for flow k becomes arbitrarily large with an infinitesimal price increase for flow ξ.

Table 7.2: Cross elasticity of demand taxonomy

We now formalise these ideas. First, we note that (7.17) can be expressed in matrix notation, so that

$$\begin{bmatrix} \log B_1 \\ \log B_2 \\ \log B_3 \\ \vdots \end{bmatrix} = \begin{bmatrix} \eta_0^1 \\ \eta_0^2 \\ \eta_0^3 \\ \vdots \end{bmatrix} + \begin{bmatrix} \eta_1^1 & \eta_2^1 & \eta_3^1 & \cdots \\ \eta_1^2 & \eta_2^2 & \eta_3^2 & \cdots \\ \eta_1^3 & \eta_2^3 & \eta_3^3 & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix} \begin{bmatrix} \log p_1 \\ \log p_2 \\ \log p_2 \\ \vdots \end{bmatrix},$$

and if we include the independent term η_0^k in the matrix of elasticities, we see that

$$\begin{bmatrix} \log B_1 \\ \log B_2 \\ \log B_3 \\ \vdots \end{bmatrix} = \begin{bmatrix} \eta_0^1 & \eta_1^1 & \eta_2^1 & \eta_3^1 & \cdots \\ \eta_0^2 & \eta_1^2 & \eta_2^2 & \eta_3^2 & \cdots \\ \eta_0^2 & \eta_1^3 & \eta_2^3 & \eta_3^3 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix} \begin{bmatrix} 1 \\ \log p_1 \\ \log p_2 \\ \log p_2 \\ \vdots \end{bmatrix}.$$

We will denote the matrix of elasticities as N, so that $N = [\eta_\xi^k]$. Furthermore, we shall denote the k-th row vector of N as N_k.

The best estimate for each of the N_k corresponding to a given flow k will be found independently by least squares minimisation. To achieve this, we require a set of measurements that give us the bandwidth that the ESP aggregate allocated to flow k, and the corresponding price vector that induced it. If we denote as B_k^t the t-th data point for B_k and p_ξ^t as the ξ-th component of the t-th price vector data point, we can produce two basic measurement matrices for the estimation of N_k. The first one relates to

observed demand, and is

$$\mathsf{T}_k^B = \begin{bmatrix} \log B_k^1 \\ \log B_k^2 \\ \log B_k^3 \\ \vdots \end{bmatrix};$$

the second one relates to the price vectors that elicited that demand, and is

$$\mathsf{T}^p = \begin{bmatrix} 1 & \log p_1^1 & \log p_2^1 & \log p_3^1 & \cdots \\ 1 & \log p_1^2 & \log p_2^2 & \log p_3^2 & \cdots \\ 1 & \log p_1^3 & \log p_2^3 & \log p_3^3 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix}.$$

Having defined T_k^B and T^p, we can state the basic optimisation problem for the elasticity-based demand model. We seek the N_k that best relates (in the squared-error sense) the observed traffic demands in terms of the supplied price vectors. Thus, the problem formally reads as

$$\underset{N_k'}{\text{Minimise :}} \quad ||\mathsf{T}^p N_k' - \mathsf{T}_k^B||_2. \tag{7.18}$$

where $||\cdot||_2$ denotes the usual Euclidean vector norm, and we solve for the transpose of N_k in order to deal with a column vector rather than a row vector. For our simulations, we solve the optimisation problem (7.18) by simply making use of the Moore-Penrose pseudoinverse [251] of T^p, which immediately yields the desired result when premultiplied by T_k^B.

7.3.2 Simulation Experiments

In order to test the elasticity-based model we have just discussed, we calculate the aggregate traffic matrix of a set of ESPs, each using the analytical model of Section 7.2 to allocate its own traffic matrix. Some results of this are presented in Figure 7.3.2 and discussed below.

The simulation was organised in *runs*, with each run consisting of the generation of 1000 random price vectors and the calculation of the corresponding traffic allocation by each ESP. Each of these pairs of price vector and aggregate traffic matrices will be called a *data point*.

The regression framework detailed in the previous section is not applied to all 1000 data points of each run, however - only to the first 400. For the other 600, aggregate demand is estimated by using the η_ξ^k coefficients calculated on the basis of the first 400.

The system is assumed to have converged from the 230-th data point onwards, so these values (data points between 231 and 1000) are used to measure the accuracy of the regression/estimation technique. This is done by generating a new random price vector, obtaining the aggregate demand induced by it, and comparing it with the aggregate demand predicted by the η_ξ^k matrix estimated on the basis of current data. The relative squared error differences are kept, and this process repeated 100 times.

In each of these runs, a system with 15 ESPs and 15 sites deployed over a single ISP is simulated. The parameters used by the simulation are drawn from random distributions as indicated in Table 7.3, and are recalculated with each one of the 100 simulation repeats.

Figure 7.3(a) shows the relative squared error in $\hat{B}$, the estimation of B given a random price vector p, for data points 230-1000. The relative squared error is defined as

$$E_{rel} = \frac{||\hat{B} - B||_2}{||B||_2}.$$

Parameter	Distribution	Mean	Standard Deviation
β_i	Uniform	30.093	2.8957
α_i	Uniform	.25	0.14434
γ_i	Uniform	.35	0.20207
δ_i	Uniform	.30	0.17321
q_{ji}	Uniform	.65	0.20207
p_{ji}	Uniform	525	101.04

Table 7.3: Parameter distributions for elasticity-based model

As evident from Figure 7.3(a), around 90% of all observations after the system have converge have errors that are within .2% of the true aggregate ESP bandwidth, as allocated by (7.12) at each ESP. Clearly, (7.17) can closely replicate the behaviour of the ESP aggregate.

It is interesting to discuss the convergence properties of the least squares model of the previous section as the number of data points in it increases. Figure 7.3(b) shows the relative square error that was calculated for a test ISP aggregate, as a function of the number of data points generated in the simulation run. The first vertical line (in a dot-dash motif) shows the point from which the system is considered to be stable, and therefore, from which statistics for Figure 7.3(a) are taken. The second line (in dashed motif) shows the last data item to be used to update the elasticity matrix. After that point, the coefficients η_ξ^k are no longer updated.

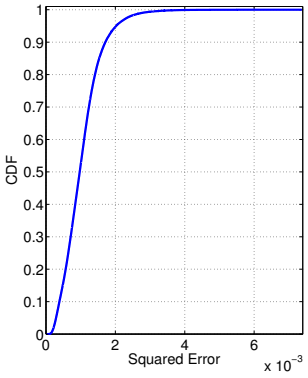
The large decrease in relative error in 7.3(b) around 210 data points is a consequence of the matrix T_k^p becoming full rank, and the MATLAB pseudoinverse code updating its algorithm accordingly (it is due to this that a value of 230 was chosen as the stability boundary for the collection of relative error statistics for Figure 7.3(a)).

The second discontinuity visible in 7.3(b), at 400 data points, is a consequence of the η_ξ^k values being no longer updated by the regression process. Consequently, relative errors obtained for all data points after this one use the same values of η_ξ^k, calculated on the basis of the first 400 data points only.

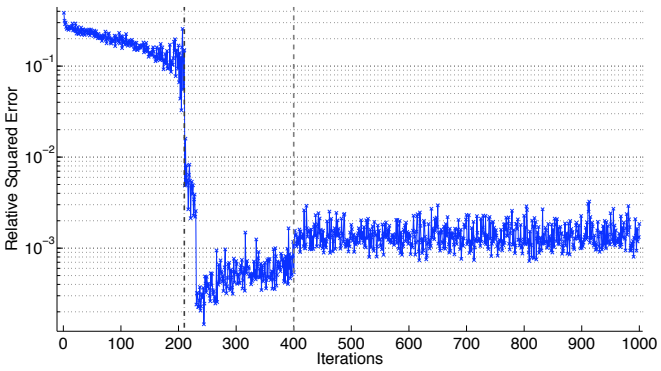
Figure 7.3(c) shows the matrix N, having removed the column which represents the constant term η_0^k. As visible from the figure colourbar on the right, most values are small and positive, with the exception of values on the diagonal which are large and negative (we explore those later). There is obvious structure in this matrix, particularly a high degree of symmetry. Exploiting this structure remains as future work.

In Figure 7.3(d) we see the CDF of the cross elasticity of demand, which shows a very large number of values near zero with a small spread to be expected from noise. If these elements are dropped, the matrix becomes sparse and can be efficiently stored and manipulated. Further, the subplot in Figure 7.3(d) shows a detail of the CDF between the two points shown, and it is evident that the small number of elasticities with large values are all positive (Figure 7.3(d) only shows the cross elasticity of demand and thus omits the diagonal of the matrix). This means that, under the model of Section 7.2, all flows behave as inelastic *substitutes* for each other. This is supported by Figure 7.3(e), which shows the price elasticity of demand of all flows (the diagonal in Figure 7.3(d), plus the zero diagonals) is always negative and greater than one in absolute value, implying that flow demand is elastic with price. This is not unexpected, given that ESPs can allocate their traffic freely among ISPs and, if price increases in one of them, they can easily reallocate it to others. This confirms the findings presented in Section 7.2.3: since flow demand behaves elastically, small increases in price will trigger large decreases in demand ($\eta_k^k < -1$). On the other hand, cross elasticities of demand are all inelastic but positive $0 < \eta_\xi^k < 1$, meaning that flows only imperfectly substitute each other, and having a cheap, good quality link is always better than not having it.

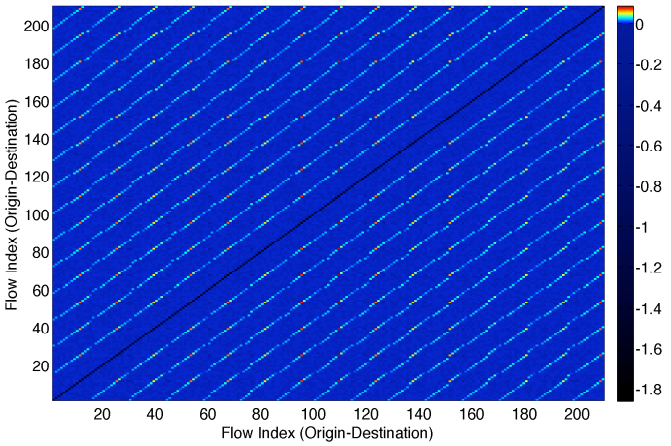
Finally, with respect to Figure 7.3(f), η_0^k is always positive, and its magnitude gives a partial view of the average level of traffic over an overlay link.



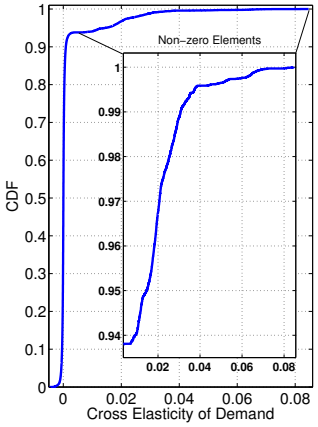
(a) CDF of the relative squared errors in the estimation for B_k



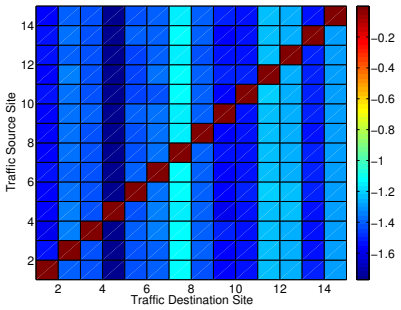
(b) Example of the relative square error behaviour as regression data increases



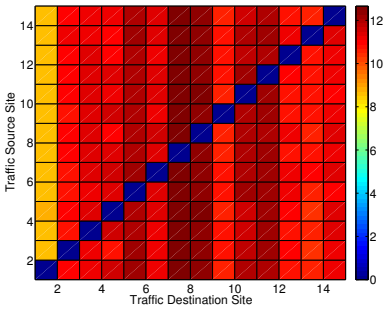
(c) Flow cross elasticity of demand matrix η_ξ^k



(d) Flow cross elasticity of demand CDF



(e) Flow price elasticity of demand vector η_k^k in matrix form



(f) Total demand factor vector η_0^k in matrix form

Figure 7.3: Estimating the parameters of the elasticity-based model

7.3.3 Using the Elasticity-based Model for Optimisation

One of the reasons an ISP might be interested in fitting a demand model would be to optimise its own prices, in order achieve maximum profit. We now turn to this issue.

As is common with some macroeconomic analyses, we assume that ESPs and ISPs can operate over two disjoint time horizons. In the *short-run* horizon the strategies of all players are constrained, with

only those strategies implementable in real time being available. On the *long-run* horizon all players can deploy all strategies, and often the convergence of short-run dynamics is assumed. We will focus on the short-run analysis, and as a result, we consider that the ISP will face *constant* per-link unit-bandwidth transmission capacity costs c_l, as well as constant per-node unit-bandwidth processing costs c_n.

Using the P4P/ALTO interface, the ISP will expose a set of origin-destination unit-bandwidth prices p_{ji} and the ESPs will behave as price takers, responding with a set of end-to-end traffic demands to be paid at such prices. We assume that they will do this by finding the end-to-end traffic demands that maximise their utility, taking the prices given by the ISP as constant (thus, we assume that the demand for ISP traffic from ESPs is *Marshallian* [144]).

For this simplified model, we treat the ISP as a monopolistic provider (this situation might be closer in spirit to the current state of affairs, where multihoming protocols have not been commonly deployed yet). Therefore, the pricing determination problem for the ISP amounts to finding the price vector p_ξ that maximises ISP profit, taking into account that B_ξ is an unknown function of p_ξ and has been estimated using the procedure detailed in the first part of this section.

Since we assume that the ISP gives the same price to every ESP, we have for the ISP profit P that

$$P = \sum_{\xi \in \mathsf{L}} p_\xi B_\xi - \sum_{l \in \mathscr{L}} c_l w_l - \sum_{n \in \mathscr{N}} c_n w_n, \tag{7.19}$$

where c_l is the unit cost of the underlying link l, w_l the link bandwidth used by ESP traffic on link l, c_n is the unit cost per computation unit on a given ISP router n, and w_n is the amount of computation units which router n will require to process all the ESP traffic going through it.

We continue the development of the model by expressing the cost terms of (7.19) in terms of the end to end flows ξ. To this end, we note that the mapping from origin-destination flow volumes to traffic volumes on any given link l is given by the *routing matrix* R^L, where the (l, ξ) element $R^L_{l\xi}$ is 1 if origin-destination flow ξ goes through link l. The mapping from origin-destination flows to the amount of traffic on any given node n is given by the *node routing matrix* R^N, where the (n, ξ) element $R^N_{n\xi}$ is 1 if origin-destination flow ξ goes through node n.

Since load patterns in links and routers tend to be bursty, the ISP might need to provision raw infrastructure capacities w_l and w_n such that the level of utilisation by ESP traffic in any node or link does not exceed a given maximum. If ρ_N is the maximum router utilisation for ESP traffic and ρ_L is the equivalent maximum utilisation for ESP traffic on links, we have the following utilisation restrictions for the ISP:

$$\sum_{\xi \in \mathsf{L}} R^N_{n\xi} B_\xi \leq \rho_N w_n$$
$$\sum_{\xi \in \mathsf{L}} R^L_{l\xi} B_\xi \leq \rho_L w_l.$$

If we assume that these two conditions are always binding (which would be equivalent to assuming that the ISP behaves as a cost minimiser), we have for the profit of the ISP that

$$P = \sum_{\xi \in \mathsf{L}} p_\xi B_\xi - \sum_{l \in \mathscr{L}} \frac{1}{\rho_L} c_l \sum_{\xi \in \mathsf{L}} R^L_{l\xi} B_\xi - \sum_{n \in \mathscr{N}} \frac{1}{\rho_N} c_n \sum_{\xi \in \mathsf{L}} R^N_{n\xi} B_\xi,$$

and by reversing the summation orders of the flows ξ and the routers and links n and l, we have that

$$P = \sum_{\xi \in \mathsf{L}} p_\xi B_\xi - \sum_{\xi \in \mathsf{L}} \left(\sum_{l \in \mathscr{L}} \frac{1}{\rho_L} c_l R^L_{l\xi} + \sum_{n \in \mathscr{N}} \frac{1}{\rho_N} c_n R^N_{n\xi} \right) B_\xi. \tag{7.20}$$

$P \in \mathbb{R}_{\geq 0}$	Profit for the ISP applying the model
$p_\xi \in \mathbb{R}_{\geq 0}$	Per-unit bandwidth price for flow ξ for the ISP applying the model
$B_\xi \in \mathbb{R}_{\geq 0}$	Flow volume for flow ξ over the ISP applying the model
ϕ_ξ	Per-unit bandwidth cost for flow ξ for the ISP applying the model
η_0^k	Magnitude parameter for the price-demand model of flow k
η_ξ^k	Cross elasticity of demand between flow k with respect to flow ξ for the ISP applying the model
$\mathscr{L}$	Set of all infrastructure links for the ISP applying the model
$\mathscr{N}$	Set of all infrastructure routers for the ISP applying the model
L	Set of all site-to-site flows for the ISP applying the model

Table 7.4: ISP profit maximisation model notation

The magnitude in parenthesis involving c_l and c_n represents the unit costs associated with a particular end-to-end flow ξ, and it will remain constant over the short term horizon. Thus, we can define the *per-unit infrastructure bandwidth cost* ϕ_ξ so that

$$\phi_\xi = \frac{1}{\rho_L} \sum_{l \in \mathscr{L}} c_l R_{l\xi}^L + \frac{1}{\rho_N} \sum_{n \in \mathscr{N}} c_n R_{n\xi}^N, \tag{7.21}$$

and substituting (7.21) in (7.20), we have that

$$P = \sum_{\xi \in \mathsf{L}} (p_\xi - \phi_\xi) B_\xi. \tag{7.22}$$

Having obtained a model for B_ξ as shown in the first part of Section 7.3, the ISP can define its optimisation problem as follows (variable definitions can be found in Table 7.4):

$$\begin{aligned} \text{Maximise:} \quad & P = \sum_{\xi \in \mathsf{L}} (p_\xi - \phi_\xi) B_\xi \\ \text{Subject to:} \quad & \log B_k = \eta_0^k + \sum_{\xi \in \mathsf{L}} \eta_\xi^k \log(p_\xi), \end{aligned}$$

where there is one restriction for each end-to-end flow $k \in \mathsf{L}$. From this we can construct the Lagrangian

$$\mathcal{L}_I = \sum_{\xi \in \mathsf{L}} (p_\xi - \phi_\xi) B_\xi + \sum_{k \in \mathsf{L}} \lambda_k \left(\log B_k - \eta_0^k - \sum_{\xi \in \mathsf{L}} \eta_\xi^k \log(p_\xi) \right).$$

We will treat p_ξ, B_ξ and λ_ξ as variables, and thus we have a set of $3|\mathsf{L}|$ equations in $3|\mathsf{L}|$ unknowns. By differentiating $\mathcal{L}$ with respect to p_g and assuming $p_\xi > 0 \, \forall \, \xi \in \mathsf{L}$, we obtain the following system of equations:

$$p_g B_g - \sum_{k \in \mathsf{L}} \lambda_k \eta_k^g = 0 \quad \forall g \in \mathsf{L},$$

which can be expressed in matrix form, so that

$$\begin{bmatrix} p_1B_1 \\ p_2B_2 \\ p_3B_3 \\ \vdots \end{bmatrix} - \begin{bmatrix} \eta_1^1 & \eta_2^1 & \eta_3^1 & \cdots \\ \eta_1^2 & \eta_2^2 & \eta_3^2 & \cdots \\ \eta_1^3 & \eta_2^3 & \eta_3^3 & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix} \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \lambda_3 \\ \vdots \end{bmatrix} = 0. \tag{7.23}$$

By differentiating $\mathcal{L}_I$ with respect to B_g and assuming $B_\xi > 0$, we obtain the system of equations

$$(p_g - \phi_g)B_g + \lambda_g = 0 \quad \forall g \in \mathsf{L},$$

which can again, be expressed in matrix form so that

$$\begin{bmatrix} p_1B_1 \\ p_2B_2 \\ p_3B_3 \\ \vdots \end{bmatrix} - \begin{bmatrix} \phi_1B_1 \\ \phi_2B_2 \\ \phi_3B_3 \\ \vdots \end{bmatrix} + \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \lambda_3 \\ \vdots \end{bmatrix} = 0. \tag{7.24}$$

If we denote the matrix of η_k^g as N and substitute (7.24) in (7.23), we have that

$$\begin{bmatrix} p_1B_1 \\ p_2B_2 \\ p_3B_3 \\ \vdots \end{bmatrix} - N \left(\begin{bmatrix} \phi_1B_1 \\ \phi_2B_2 \\ \phi_3B_3 \\ \vdots \end{bmatrix} - \begin{bmatrix} p_1B_1 \\ p_2B_2 \\ p_3B_3 \\ \vdots \end{bmatrix} \right) = 0,$$

which simplifies to

$$[I + N] \begin{bmatrix} p_1B_1 \\ p_2B_2 \\ p_3B_3 \\ \vdots \end{bmatrix} = N \begin{bmatrix} \phi_1B_1 \\ \phi_2B_2 \\ \phi_3B_3 \\ \vdots \end{bmatrix},$$

and if we define the matrix $\Theta = [I + N]^{-1}N$, we have that

$$\begin{bmatrix} p_1B_1 \\ p_2B_2 \\ p_3B_3 \\ \vdots \end{bmatrix} = \Theta \begin{bmatrix} \phi_1B_1 \\ \phi_2B_2 \\ \phi_3B_3 \\ \vdots \end{bmatrix}. \tag{7.25}$$

If we assume that $p_\xi > 0$, $B_\xi > 0$ and that Θ is of full rank ($\det \Theta > 0$, so that its kernel is null), we can apply the $\log$ function at both sides of the equal sign for each element. Thus doing, we obtain that

$$\begin{bmatrix} \log p_1 + \log B_1 \\ \log p_2 + \log B_2 \\ \log p_3 + \log B_3 \\ \vdots \end{bmatrix} = \begin{bmatrix} \log \sum_{\xi \in \mathsf{L}} \theta_{1\xi}\phi_\xi B_\xi \\ \log \sum_{\xi \in \mathsf{L}} \theta_{2\xi}\phi_\xi B_\xi \\ \log \sum_{\xi \in \mathsf{L}} \theta_{3\xi}\phi_\xi B_\xi \\ \vdots \end{bmatrix}. \tag{7.26}$$

However, from (7.17) we can see that

$$\begin{bmatrix} \log B_1 \\ \log B_2 \\ \log B_3 \\ \vdots \end{bmatrix} = \begin{bmatrix} \eta_0^1 \\ \eta_0^2 \\ \eta_0^3 \\ \vdots \end{bmatrix} + N \begin{bmatrix} \log p_1 \\ \log p_2 \\ \log p_2 \\ \vdots \end{bmatrix},$$

and thus, that

$$\begin{bmatrix} \log p_1 \\ \log p_2 \\ \log p_3 \\ \vdots \end{bmatrix} = N^{-1} \left(\begin{bmatrix} \log B_1 \\ \log B_2 \\ \log B_2 \\ \vdots \end{bmatrix} - \begin{bmatrix} \eta_0^1 \\ \eta_0^2 \\ \eta_0^3 \\ \vdots \end{bmatrix} \right).$$

If we substitute this last expression in (7.26) and we denote the vector with the η_0^ξ as $\bar{\eta}_0$, we obtain that:

$$\left[N^{-1} + I\right] \begin{bmatrix} \log B_1 \\ \log B_2 \\ \log B_3 \\ \vdots \end{bmatrix} - N^{-1}\bar{\eta}_0 = \begin{bmatrix} \log \sum_{\xi \in \mathsf{L}} \theta_{1\xi}\phi_\xi B_\xi \\ \log \sum_{\xi \in \mathsf{L}} \theta_{2\xi}\phi_\xi B_\xi \\ \log \sum_{\xi \in \mathsf{L}} \theta_{3\xi}\phi_\xi B_\xi \\ \vdots \end{bmatrix},$$

and if we define $\Psi = [N^{-1} + I]$ and $\bar{\epsilon}_0 = N^{-1}\bar{\eta}_0$, we obtain the system of nonlinear equations

$$\Psi \begin{bmatrix} \log B_1 \\ \log B_2 \\ \log B_3 \\ \vdots \end{bmatrix} - \begin{bmatrix} \log \sum_{\xi \in \mathsf{L}} \theta_{1\xi}\phi_\xi B_\xi \\ \log \sum_{\xi \in \mathsf{L}} \theta_{2\xi}\phi_\xi B_\xi \\ \log \sum_{\xi \in \mathsf{L}} \theta_{3\xi}\phi_\xi B_\xi \\ \vdots \end{bmatrix} = \bar{\epsilon}_0. \tag{7.27}$$

We can obtain the best solution to the system of equations (7.27) over $\mathbb{R}_{\geq 0}$, in the least-squares sense, using the Levenberg-Marquardt algorithm[218]. Once a set of B_ξ has been determined, we find p_ξ from (7.25) so that

$$\begin{bmatrix} p_1 \\ p_2 \\ p_3 \\ \vdots \end{bmatrix} = \begin{bmatrix} B_1 & & & \\ & B_2 & & \\ & & B_3 & \\ & & & \ddots \end{bmatrix}^{-1} \Theta \begin{bmatrix} \phi_\xi B_1 \\ \phi_\xi B_2 \\ \phi_\xi B_3 \\ \vdots \end{bmatrix}.$$

If needed, the Lagrange multipliers can be found from (7.23) such that

$$\begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \lambda_3 \\ \vdots \end{bmatrix} = N^{-1} \begin{bmatrix} p_1 B_1 \\ p_2 B_2 \\ p_3 B_3 \\ \vdots \end{bmatrix}.$$

7.4 Discussion of Potential Improvements

In this chapter we have shown two different ways of estimating the response of one or more mutihomed ALTO-managed overlays to changing prices.

The first of the models, presented in Section 7.2, is constructive, but requires the estimation of parameters from competing ISPs. Truthful revelation is of course a concern in that case; even if ISPs

deployed overlay sites which purchased services from their competitors as if they were conventional members of an ESP, these sites could be given special treatment in order to guide the estimation of the model parameters to any desirable outcome. Therefore, the development of end-to-end quality measurement tools that do not rely on the cooperation of intervening ISPs might be a precondition to any practical proposal in this respect. Of course, network tomography techniques [303, 272, 301] might be of relevance for this issue. Of course, the extension of the model so that it considers traffic volume constraints would be a fruitful avenue for further development for this model.

The second model, although structurally much simpler, presents significant numerical challenges, in particular the matrix inversion required to solve (7.18) and the solution of the potentially very large nonlinear system of equations (7.27) if used for profit maximisation. Research into how to perform these functions efficiently, or into more efficient alternative models, remains an important topic deserving of further study.

8 Conclusions

Although the study of incentive mechanisms is a very mature field of study, branching out from fundamental problems of both economics and computer science, there is still ample space for contributions in the area. The main reason for this is that there is no "silver bullet" for the design of incentive mechanisms: the variety in their theory and implementation mirrors that one present in the design of peer-to-peer overlays themselves, and the tradeoffs that guided the engineering decisions for each one of them.

In this context, the contributions presented in this thesis do not represent the "best" solution to the incentives problem in massively distributed systems, since in this case "best" is an ill defined term. Different objectives will yield different resource allocation strategies, which will have various degrees of alignment with those required for incentive mechanism design.

The objective of this thesis is to provide novel techniques that allow incentive mechanism designers to make efficient tradeoffs between overlay optimality and incentive mechanism resource allocation, particularly in the context of QoS overlays. Moreover, since the contributions in this thesis arise from taking situations where incentive tensions are present, and then proposing either an algorithm to resolve them, or a model that can be used to study them in greater depth, they are relevant both in theoretical or practical incentive mechanism design situations.

8.1 Summary of Contributions

We now summarise the main contributions of this thesis, and discuss the degree to which they satisfy the main properties defined in Chapter 1. These properties are restated again here, for the benefit of the reader.

Property 1 (Fundamental Property of Incentive Mechanisms)**.** *An incentive mechanism makes the level of access to overlay resources, including the service quality obtained from the system, contingent on the amount and quality of the resource contributions that peers have made to it.*

Property 2 (Deferred Indirect Reciprocity)**.** *The incentive mechanism is designed to allow peers to contribute services to a set of peers at a given time, and obtain resources corresponding to those contri-*

butions from a different set of peers at a different time, with no previous direct reciprocity requirement between them.

Property 3 (Resistance to Identity Attacks)**.** *The incentive mechanism is designed to be resistant against sybil and whitewashing attacks, by ensuring that the profit that peers can obtain from assuming multiple identities is limited.*

Property 4 (Resistance to Untruthful Peers)**.** *The incentive mechanism is designed to be resistant against untruthful peers, by ensuring that peers are unable to profit from modifying or discarding the protocol messages sent by other peers.*

Property 5 (Decentralisation)**.** *The incentive mechanism is designed to operate without need for centralised or federated infrastructure relying on agreements between administrative domains.*

Property 6 (Private Storage)**.** *The incentive mechanism is designed so that peers store a record only of the contributions that they have given to other peers and received from other peers, not the contributions between third parties.*

Property 7 (Efficiency)**.** *The system maximises the aggregate value that the peers obtain from it, by giving priority to those service requests that bring the most benefit to other peers.*

Property 8 (Load-Awareness)**.** *The cost of obtaining services from a peer increases as its usage level approaches its capacity.*

Property 9 (Preference Freedom)**.** *Each peer is able to define its own set of preferences over possible resource allocations. Equivalently, the benefit that a given peer obtains from the system is only decided by that peer.*

Property 10 (Network-Based Service Quality Model)**.** *The incentive mechanism includes a model of the influence of network conditions on the service quality received by a peer, as a function of the behaviour of its serving peer and the prevailing network conditions.*

Property 11 (Resistance to Hidden Action)**.** *The incentive mechanism provides an incentive to serving peers to deliver their advertised service quality accurately, even if non-compliance could be in theory be attributed to the prevailing network conditions.*

Property 12 (Flexibility)**.** *The preferences model for the managed overlay can be applied to a situation where there is only one overlay and multiple ISPs, multiple overlays and a single ISP, and both multiple overlays and ISPs.*

Property 13 (Ease of Computation)**.** *The preferences model for the managed overlay should be scalable, so that situations with thousands of network endpoints and hundreds of both ISPs and overlays can be considered.*

Property 14 (Parsimony)**.** *The preferences model for the managed overlay should make use of the minimum number of parameters.*

8.1.1 *PledgeRoute* and Indirect Reciprocity

The first contribution of this thesis, presented in Chapter 4, was *PledgeRoute*: a system that enables overlay network peers to contribute resources to a set of peers and use these contributions to obtain resources from a different set of peers at a different time. The system is distinguished, however, by being resistant to sybil, slandering and whitewashing attacks. When applied in conjunction with well known

indirect reciprocity techniques, *PledgeRoute* operates as an incentive mechanism by ensuring higher service quality for those peers that contribute more resources to the overlay.

We achieved this by analysing the contribution transfer operation using a routing algebra with sybil-proofness properties similar to those of maximum flow. To make the system scalable, we proposed a topology sampling algorithm that extracts sparse subgraphs with high contribution transfer potential.

Since the contribution transfer operation tends to reduce the absolute amount of contributions in the network, we proposed a contribution seeding algorithm based on the detection of credit cycles, which can the be used to infuse the network with social capital through virtual contributions between peers.

Finally, we presented incentive schemes that work alongside the aforementioned techniques in order to ensure the correct operation of the protocol in the presence of strategic peers. When coupled with simple reciprocity policies, *PledgeRoute* allows contributing peers to obtain higher service quality than freeloaders, thus becoming an incentive mechanism suitable for peer-to-peer networks.

In Chapter 1 we expressed our interest on Properties 1, 2, 6, 3, 4 and 5 as a motivation for the contributions in Chapter 4. We now discuss the extent to which the system we proposed actually has these properties.

By design, the system presented satisfies Properties 2, 6, 4 and 5. Although this is very clear for Properties 2, 6 and 4, it is unclear that the interest digraph $\mathcal{H}$ could be effectively maintained in a distributed fashion. This, fortunately, is irrelevant to the fact that the system will be able to improve upon *whatever* content location information that is available; no requirement on the fidelity of $\mathcal{H}$ is needed. Furthermore, the work in both DHTs [210] and epidemic search algorithms [297, 177] suggests that a fully distributed search overlay can provide a good approximation of $\mathcal{H}$. In light of this, we grant Property 5 to *PledgeRoute* and its accessory reciprocity policies.

Finally, with respect to the fundamental Property 1, it is clear that a peer that operates predominantly in the reciprocative regime will experience a better QoS than a peer that operates predominantly in the altruistic regime. The precise quality experienced will be a function of the utilisation level of the peers from which service is being requested, since if these peers are idle, no service degradation is incurred (there are no requests to give priority over the altruistic ones). However, by imposing a limit on the amount of resources that can be obtained in the altruistic regime (the *safe credit margin*), an incentive is created for peers to switch to the reciprocative regime irrespectively of this.

With respect to the other properties, the system proposed in Chapter 4 is not sufficiently nuanced to exhibit any of them - although it may be argued that it could be trivially modified to exhibit some measure of load awareness, by informing peers of the ratio of altruistic and reciprocative requests that they are serving.

8.1.2 Auctions, Quality and Resource Allocation

The second contribution presented in this thesis, in Chapter 5, was a swarming protocol for QoS overlays. This swarming protocol relied on the calculation of some statistics for multi-item Vickrey auctions, which were provided in closed form. In particular, the expected number of won bids, the standard deviation of the number of won bids and the expected utility as a function of the bid value distribution at the auctioneer peer were presented. These statistics were used as input of a simple, greedy peer selection algorithm that attempts to maximise the expected utility for each sent bid. Once the auction bidders are determined, each one is mapped to a QoS class depending on the balance of its past contribution accounts, thus giving better service to those peers who have contributed more to the overlay. This generic model is then applied to a specific case: delay sensitive chunk swarming. This is done by providing valuation and quality mapping functions that encode the engineering objectives of the chunk swarming system, in this case minimising stream lag from the peercaster. The swarming protocol was then tested in different peer delay distributions, with different peercaster positions, and key performance indicators were measured.

The design objectives for the proposed system were Properties 7, 8 and 9. We now address them in turn, and discuss how well does the proposed system achieve them.

With regards to efficiency (Property 7), the fact that the Vickrey auction is efficient is enough to ensure that if a peer is denied service, the utility lost by this is more than made up with the utility obtained by the peers that did win the auction. This means that the system decomposes itself in a series of Pareto-improving episodes, always tending towards a higher aggregate utility - and the system proposed exhibits Property 7. Moreover, through the expectation equation (5.6), the system can provide the information necessary to achieve Property 8. Since the system is expressly designed to allow per-peer valuation equations, it satisfies Property 9 by construction.

With respect to the other properties, it is well known that the Vickrey auction is vulnerable to manipulation by a coalition of losing bidders [242] and thus the auction-based system loses Property 3 from *PledgeRoute*. However, since the system does not include the additional contract-enforcing policies presented in addition to *PledgeRoute* in Chapter 4, it is vulnerable to moral hazard in contract delivery (see Chapter 6 and Appendix A.4). On the other hand, the system maintains Properties 2, 6, 4 and 5.

8.1.3 Unobservability, QoS Contracts and Hidden Action

The third contribution presented in this thesis can be found in Chapter 6, where we presented a model for the design and verification of QoS contracts in peer-to-peer overlays implemented over best-effort networks, and applied it to a specific case: a mesh-based, pull-oriented streaming system with low delay requirements. We treated each transfer of a delay-sensitive chunk as a *hidden action* situation where server peers can force negative externalities upon client peers, and we proposed a contract drafting procedure that allowed peers to deploy incentives to counter hidden action. This is an early step in the analytic treatment of QoS as a strategic peer behaviour, and towards a general theory of service level agreements among strategic peers. We showed that the construction of optimal contracts through the use of the *principal-agent* model can be used to ensure predictable QoS behaviour, even with utility-maximising rational peers whose service differentiation processes are unobservable. This was achieved by requiring the server peer to accept part of the externality that would have been exclusively endured by the client. Additionally, we also presented its application to a chunk transfer scenario with transaction time as the main QoS indicator. For this case, we presented and analytic model for the model parameters which can be constructed from truthfully revealed values and the analysis of observable network conditions.

Finally, we studied the enforcement of these contracts as a repeated game, and showed that a centralised entity can define the resulting Nash equilibrium to be a cooperative one by specifying the minimum probability of pairwise repeated interaction (discount parameter).

This contribution addressed a very well defined problem which, while not exclusive to QoS-overlays, has received remarkably little attention from the research community in general [330, 120, 229, 118]. The hidden action model in Chapter 6 was designed to exhibit Properties 11 and 10, which is explicitly achieved through Theorem 6.1 and equation (6.15). Moreover, from its use of *PledgeRoute* as a payment infrastructure the system maintains Properties 2, 6 and 4. Finally, whether the system satisfies Property 5 or not depends on the implementation of the market system $\mathcal{M}$: a distributed marketplace implies that the system exhibits Property 5. The system does not, however, by itself exhibit Property 3, as it simply assumes an external trusted identity management system.

8.1.4 Overlay Utility Optimisation, Prices and the Elasticity of Demand

The final contribution of this thesis can be found in Chapter 7, where we presented a general model of the incentives of managed overlays based on a simple utility function that satisfies many of the common intuitions regarding the behaviour and preferences of a service provider that relies on edge-deployed infrastructure to provide QoS-aware services. The model is solved analytically, both for a budget-free

case and one with a maximum budget condition. The model is explored numerically, and is showed to exhibit flow substitution properties: ESPs[1] will adapt their consumption patterns, using more bandwidth on cheap overlay links than on expensive ones (if both have the same quality). Given some practical limitations on fitting free parameters of this model, another one was proposed that operates in a purely numerical fashion, minimising the expected error of a constant-elasticity demand function similar to the *Cobb-Douglas* production function [83]. This function is shown to have high accuracy and great flexibility in recovering the behaviour of an aggregate of ESPs. This model is then used analytically to maximise ISP profit.

Not being an incentive mechanism but, rather, a model for the behaviour of an economic actor, the model in Chapter 7 will not exhibit many of the properties that we have been discussing. We focus on the analysis of Properties 12, 13 and 14, which were the stated engineering objectives for this contribution.

Of course, the detailed analytic model of Section 7.2 exhibits Property 12 by design. The numerical approach of Section 7.3, however, only includes the price vector of a single ISP and might be insufficiently detailed to account for price competition between ISPs (although this might change by considering an expanded matrix $\eta^k_{\xi\zeta}$ that considers the cross elasticity of demand between different flows in different ISPs). Understandably, the greater detail of the analytic model of Section 7.2 means that much lower computation power is needed for its evaluation, and thus we say that it exhibits Property 13. This is not as clear in the regression model of Section 7.3, even if we choose to avoid the pseudoinverse calculation entirely and invert the matrix using iterative methods with better sparsity properties.

Finally, it is evident that the analytic model of Section 7.2 is much more efficient in its use of parameters than the model of Section 7.3, which then would be considered less parsimonious. This is confirmed by the fact that many of the coefficients in the elasticity matrix N are close to zero, and thus essentially useless in the definition of the demand dynamics of the ESP aggregate.

8.2 Further Work

Although the work in this thesis is self-contained, it can be easily used as a starting point for continued research. We now discuss some possible avenues for further work.

Regarding sybilproof incentive mechanisms (such as *PledgeRoute* in Chapter 4), a full mathematical theory of sybilproofness is still an open research question. Although [71] and [262] have provided elegant mathematical descriptions of sybilproofness, a fully algebraic characterisation (close in spirit to what [146] has achieved for routing) could be potentially used, given a suitable metalanguage for the description of peer preferences and reciprocity algorithms, to the automatic compilation and deployment of provably sybilproof protocols, which might extend beyond QoS-overlays and into peer-to-peer auctions [136] or peer-to-peer lending [70].

The main way in which the work in Chapter 5 could be continued would be with a full characterisation of the exact proportional quality mapping algorithm of Section 5.2.2.1. The definition and experimental validation of the quality estimation function F_i, as well as the development of robust descriptions of the q_c-quality level sets $\hat{\Omega}^i(q_c)$ would then be a prerequisite not only to the solution of the proportional quality mapping problem, but to many other models that involve situations where a link between resource allocation and application layer service quality is needed. Of course, efficient numerical methods to solve the quality mapping problem with restrictions in a scalable or distributed fashion, and its application to a suitable practical example, would be of great interest[2].

Regarding the hidden action model presented in Chapter 6, many possible research avenues are available. The most straightforward one would be the extension of the principal-agent model presented to

[1] As defined in Chapter 7, an ESP is an Edge Service Provider - a managed overlay.

[2] The informational efficiency techniques from [169, 261, 168] might be of use for this.

other situations with hidden action, such as interdomain QoS routing, particularly when paying attention to revelation incentives. However, the model itself is a very simplistic one, and its generalisation to a full-fledged moral hazard principal-agent model [196] could yield important results regarding the incentives of service level agreements. Of course, doing this would require a much more sophisticated model of principal and agent risk avoidance and preference, which may be significantly different when we compare ISPs, peer-to-peer software users and CDN or managed overlay sites. A possible outcome of this work would be the development of *QoS insurance policies*, which itself could become a standard feature of service level agreements.

Finally, the experimental validation of the models presented in Chapter 7, as well as the creation of better models based on ISP data would be of great interest. However, the most promising future work stemming from this work would be the analysis of price convergence dynamics in such a system, and its relationship to optimal profit outcomes to the ISP when considering accumulated profit over time. Intuitively, the ISP wants to converge to a profit-maximising price vector as quickly as possible, but without inducing instability on the demand dynamics of the ESPs. In this case, an iterative supergradient-based approach [320] does not seem appropriate, and techniques from optimal nonlinear network control may be required [222].

Finally, this thesis has shown each one of the incentive mechanism components in Chapters 4, 5, 6 and 7 in isolation. The creation of a comprehensive, flexible incentive mechanism that integrates all the contributions of this thesis to achieve flexible and efficient QoS overlay operation that achieves high user satisfaction while being predictable to the ISP remains as one more way in which the work in this thesis could be continued.

Part III

Appendices

A

Designing Peer-to-Peer Protocols: Can the Social Sciences Help?

As has been suggested in Chapters 1 and 2, there is no shortage in the social science literature of analyses relevant to communities of autonomous agents. Here, we present a necessarily brief excerpt of this material that, although not required to understand the main contributions in this thesis, is interesting and relevant. The treatment has been focused on the analysis of peer-to-peer network protocols, and the intention is that it will help incentive mechanism designers in the requirements definition phase to set the basic objectives for their systems.

A.1 Strategic Agents

Given that human beings themselves are, in a way, autonomous peer-to-peer agents, it is unsurprising that many of the problems that arise in the analysis and design of emergent behaviours for peer-to-peer networks had already been encountered before in economics and sociology, in an ongoing attempt to answer the question of what is the aggregate behaviour of a large society composed of rational (and not so rational) agents. However, we will not attempt to do a complete mapping between these disciplines and network engineering. Instead, we shall focus on the parallels between humans and peer-to-peer elements that are quick to emerge, because they correspond to intrinsic characteristics of strategic interaction. Thus, the focus will be in fundamental behavioural elements of free agents. We now analyse the most important aspects relating to the behaviour of rational peers.

The first insight stemming from the economics of general equilibrium [144] is that distributed resource allocation is, essentially, an exercise in *information diffusion*. Since resource allocation in peer-to-peer networks depends on peer resource availability and the preferences that guide the behaviour of each peer, the knowledge required to produce a welfare-maximising resource allocation is *decentralised* over the whole network. No single peer can know the allocations that are feasible, let alone those that are optimal. Thus, distributed resource allocation requires the diffusion of peer information and the decisions taken on this basis.

The precise way in which information propagates, and how reliable it is, will depend on the behaviour of the peers. It follows, then, that a greater understanding of the behaviour of peers will be useful if we are to model them more accurately. In [283], Shneidman and Parkes propose the following taxonomy of peers in peer-to-peer systems:

- **Obedient peers** do not change their defined protocol behaviour to seek their best outcome. Thus, obedient peers will stick to determined protocols, regardless of the utility that they receive from their interaction with other peers.

- **Rational peers** use knowledge of the environment, other peers, the protocol and the underlying network substrate to maximise the expected utility they can extract from the peer-to-peer system. Rational peers can thus behave, if in their benefit, as *adversarial* peers that propagate false in-

formation, selectively modify or drop other nodes' communications, or supplant or substitute any part of the protocol with one more closely aligned with their utility maximisation strategy.

- **Irrational peers** are strategic, but act according to utility functions that are not under the control of the mechanism designer. These might be known or unknown.

- **Faulty peers** have widely varying behaviours, from complete removal from the peer-to-peer overlay (*failstop* peers) to Byzantine failures (arbitrary protocol actions). Since the behaviour of these peers is essentially random, their effect in any incentive mechanism should be minimised.

Of all these, only the first three can propagate information with any degree of accuracy, and only the first two can be directly analysed using a game theoretical formalism. Thus, we shall focus on these two categories of peers.

A.1.1 Bounded Rationality

The usual assumption in the economic or game-theoretical analysis of autonomous agent behaviour is that agents are *rational*: when faced with a particular decision, they will always know (and prefer) the strategy that maximises their expected utility. However, economic agents have limited information and limited ability to process it: In practice, this maximisation might impose large computational effort, large amounts of memory or a large amount of signalling bandwidth.

The usual way of modelling these limitations is through *bounded rationality*: agents are still expected to attempt to maximise their utility, but instead of using full-fledged optimisation mechanisms they will use simple heuristics that provide results that are "sufficiently good" (satisfy a minimum utility criterion).

Thus, boundedly rational agents always need to operate without complete information about market opportunities, with limited ability to predict the future (and adequately reason about the implications of these predictions), and with restricted capability to pre-specify optimal responses to future events. Agents cannot know everything and inevitably make mistakes, and each agent's knowledge is slanted to its immediate social or geographical environment.

A.1.2 Strategic and Opportunistic Behaviour

In peer to peer networks, every peer is normally under the administrative control of a distinct user. Thus, peers are expected to modify their behaviour to maximise the benefit they get from the system. If we define the universe of actions that can be taken by each peer as its *strategy space*, we see that peers will attempt to select strategies that maximise their utilities. We call this kind of peer behaviour *strategic*.

Sometimes, strategic peer behaviour is opportunistic: peers will attempt to take advantage of environmental conditions and the limitations of other peers in order to increase their own utility. Opportunism follows from bounded rationality plus *self-interest*. When a conflict arises between what an agent a can do to increase its own utility and what it agreed to do for another agent b to increase b's utility, a will act on its own interest, insofar as it is costly for b to discover this (b is only boundedly rational) and to punish a. It is not necessary to assume that every agent always behaves opportunistically in order to have opportunism as a transaction cost; it is sufficient that some agents act opportunistically some of the time, and that boundedly rational agents are unable to predict if any other agent will behave opportunistically or not.

A.1.3 Protocol Design with Strategic Agents

Many of the distributed systems protocols currently in use rely on information dissemination for their operation. If one considers strategic agents, protocol design becomes substantially more complicated: in that case, peers can choose to withhold or falsify information in order to modify the resource allocation process in a manner that is advantageous for them. This problem can be analysed using *Mechanism Design* [169, 304] (see Section 3.4.2). In this formulation, a peer-to-peer protocol is an instance of an economic mechanism. The strategies of the peers are modelled using a *game form*, and desired protocol outcomes are given by a *goal function*. The mechanism is thus designed to produce, given the peer strategy space, an equilibrium consistent with the goal function. In order for this to be feasible, the mechanism must place realistic information processing and exchange demands on the peers. Thus, the design of *informationally efficient* mechanisms finds a natural application in the design for peer-to-peer protocols.

A.2 Markets and Social Institutions

A.2.1 Transaction Costs and Governance

In human societies, the problem of regulating self-seeking behaviour was one of the first requirements for society growth after the development of agriculture [199]. This is because of the notion of *transaction costs*: the costs incurred in making an economic exchange. Basically, every economic transaction is dependent on the predictability of its outcome. If economic transactions have intrinsic *risks* that outweigh their potential benefit, there might not be a rational incentive to participate in them.

There are basically three kinds of transaction costs [199, 18]:

- **Search and Information Costs**: the costs of finding an appropriate product/service to satisfy a given need, and to find an appropriate trading party to obtain it from. Typical examples of this are the efforts that vendor exert in advertising or market research. In peer-to-peer systems, this amounts to bandwidth and computation needed for finding and advertising resources. In the case of delay-sensitive overlays, this represents the cost of finding, within the limited time provided by the media play-out buffer, a set of peers that have the chunks that it requires, and that can provide them reliably (at an appropriate level of QoS) during an appropriately long time interval. These operations need to be done in conditions of constant variation in network RTT, end to end throughput and peer churn.

- **Exchange Costs**: the costs of actually performing the exchange in a mutually beneficial fashion. Usually this implies that the parties must bargain until they agree on the terms of the trade and draw an appropriate contract. Typical examples of this are the effort that a vendor needs to exert in order to sell his/her wares (transporting them to the market, maintaining the shop/stall, etc.) and the effort incurred by both parties in price determination. In real-time peer-to-peer streaming systems, this amounts to the cost of actually obtaining the relevant chunks in an environment of strategic peers: bandwidth and computation related to trade conditions agreement (such as price setting), actual service delivery (download bandwidth for files, or QoS streaming for video systems) and the establishment of exchange networks (swarming scheduling algorithms).

- **Policing Costs**: the costs of preventing, detecting and punishing opportunistic behaviour even though it might be non-observable and only inferrable through transaction outcomes. In short, making sure that other party sticks to the terms of the contract. Typical examples of this are the effort that legislating bodies need to exert to create laws to punish theft and fraud, and the effort that police forces need to exert to enforce these laws. In peer-to-peer systems, an example of this is the bandwidth and computation needed for bootstrapping and maintaining reputation systems (like upload/download ratio systems in BitTorrent communities), or the withholding of reciprocation even after an adequate service has been performed. As the actions of peers can only be observed statistically, network variability can be confused with peer opportunistic behaviour. This implies that peers need to follow enforcement behaviours that are able to recover from these errors, while maintaining a reliable cooperative regime.

A.2.2 Market Governance and Social Institutions

In general, market mechanisms work well for situations that have low transaction costs. For highly costly transactions, where the market imposes too high overheads, other forms of *governance*[1] [45] might be necessary to regulate exchange.

The determinant characteristics of transaction costs are, then, according to [30, 317]:

- **Frequency**: If an agent a requires the services of an agent b very frequently, it will be beneficial to both to define streamlined procedures that can take into advantage the agents' mutual dependence. Thus, for highly frequent agent interactions, efficiency can be increased by bypassing the market and relying on non-market governance instead. An example of this would be a relationship that, developed through continuous interaction, becomes more reliant on trust management mechanisms, such as reputation systems, than on market governance.

- **Specificity**: A transaction has high specificity when the assets intended for use in it can not be easily transferred to other uses. Highly specific assets represent *sunk costs* [2]that have relatively

[1] From [18]: Governance is that separate process or certain part of management or leadership processes that makes decisions that define expectations, grant power, or verify performance. Frequently a **government** is established to administer these processes and systems.

little value beyond their use in the context of a specific transaction. It follows that, if both agents a and b engage in exchanges with high specificity, the value of this exchanges is much greater for each other than for any other arbitrary agent. Under this circumstances, efficiency can be increased by bypassing the market and relying on non-market governance instead.

- **Uncertainty**: In transactions with uncertainty, there is an essential difficulty in foreseeing the the eventualities that might occur during the course of the transaction. Clearly, one very important element in whether a transaction is uncertain or not is the length of time over which the transaction will take place. Transactions that take place instantaneously will have little uncertainty, because the risks involved in incorrectly predicting the other agent's actions and circumstances in the future is nonexistent. This is the basis for *spot markets*[3], where transactions are simplified by eliminating credit risk.

A.2.3 Information and Markets

Any resource allocation that is realised through pairwise exchange implies that both peers involved in each exchange will be in a "better" state than before the transaction (for any definition of "better" that guides peer decisions). Thus, every trade moves the system towards Pareto optimality. This, unfortunately, does not mean that the equilibrium reached by this process is especially efficient or fair. It will, however, be implementable in a distributed fashion by strategic agents.

Resource allocation through trade is not a "zero-sum game": it is a mutually beneficial activity. However, for each trade to be as optimal as possible, every peer should have a reliable, up-to-date view of all its current trading possibilities with every other peer in the system. This particular model of perfect competition, rooted in *perfect information* (where every bit of information in the system is leveraged in every economic decision, so that economic decisions are correctly valued) is not only very difficult to achieve, but also not scalable. In Hayek's words [162]:

> *The peculiar character of the problem of a rational economic order is determined precisely by the fact that the knowledge of the circumstances of which we must make use never exists in concentrated or integrated form but solely as the dispersed bits of incomplete and frequently contradictory knowledge which all the separate individuals possess. The economic problem of society is then not merely a problem of how to allocate "given" resources - if "given" is taken to mean given to a single mind which deliberately solves the problem set by these "data." It is rather a problem of how to secure the best use of resources known to any of the members of society, for ends whose relative importance only these individuals know. Or, to put it briefly,* ***it is a problem of the utilisation of knowledge which is not given to anyone in its totality***.
>
> (quote from [162]; emphasis added)

As a consequence, distributed markets with perfect information (where all participants can reliably know the strategies and actions of every other participant) tend to be very expensive and difficult to implement in practice.

A.2.4 Market Failure

If the system outcome achieved by a market solution is *inefficient*: some peers could have greater utilities without decreasing the utility of any other peer. One usual reason why markets fail is that they have excessively high transaction costs, which in turn is dependent on transaction uncertainty. Transactions that are resolved instantaneously tend to have low uncertainty. On the other hand, transactions that involve a commitment over some time have some minimum intrinsic uncertainty, w which then increases with the length of time over which the commitment is maintained. In general, uncertainty is dependent on the characteristics of the agents:

- *Bounded Rationality*: Agents that are computationally and informationally strong, and consequently more capable of acting as ideal rational agents, are less influenced by uncertainty than

[2]Sunk Costs are costs incurred in the course of a transaction, or in the production of goods and services, that cannot be recovered later on - not even by the possible sale of the asset they were used to produce. Therefore, they represent **barriers to exit**: an economic actor that has incurred high sunk costs will prefer to continue its current course of action, which legitimates the expenditure of the sunk cost, rather than choosing another course of action that would make the sunk cost a loss. This can lead to irrational agent behaviour, in the form of *loss aversion*.

[3]Spot Markets are markets in which goods are sold for cash and delivered immediately. This means that contracts are immediately effective, and there is no risk with future partner behaviour involved.

those that are more limited. Thus, all things being equal, more limited agents will exhibit greater transaction costs due to uncertainty than those agents that cannot reliably extrapolate the possible outcomes of a transactions based on their past and current experience. Smart agents, then, tend to perform better in uncertain markets than more severely limited ones. Conversely, if a given environment calls for cheap, limited agents, market-based allocation may be too costly.

- *Information Asymmetry*: Agents that know almost as much as their trading parties can make much better guesses regarding their behaviour than those agents with greater information asymmetries. Then, all things being equal, agents which suffer from larger information asymmetry will exhibit greater transaction costs due to uncertainty than those agents who can somehow learn the information that their trading parties have.

- *Opportunistic Behaviour*: Agents that can rely on their parties to uphold the commitments between them are in a much better position to predict the outcome of their transactions than agents that do not know whether their trading parties will uphold their contracts. This means that, all things being equal, agents which unable to conduct most of their interactions with trusted agents will exhibit greater transaction costs due to uncertainty than those who trade with a close knit group of trusted peers. Market-inspired resource management thus works particularly well amongst peers that have long trading histories, or that can rely on external trust bindings.

A.2.5 Social Institutions

The emergence and maintenance of social institutions in human societies can be instructive to the problem of designing distributed incentive mechanisms for strategic agents. From an evolutionary anthropology perspective, the face-to-face interaction situations of the primitive man could be modelled using an iterated *Prisoner's Dilemma* [199]. Thus, both in terms of gene and meme evolution, those social behaviours that increased human fitness in instinctively achieving high payoffs would be preferentially selected, leading to the adoption of reciprocal altruism as a stable strategy [96, 299, 107]. *Social institutions* were born as the cultural outcome of this natural selection process on the arena of self-replicating units of thought and behaviour.

Therefore, cultural evolution is a good model explaining how human markets developed, even though boundedly rational, strategic peers in environments with information asymmetry tend to have large transaction costs. In human society, markets are supplemented with social institutions that attempt to reduce transaction costs, either by regulating trade or by deterring, detecting and punishing self-seeking behaviour. These may be either formal (*law*) or informal (*morality*). The main difference between these two cases is the way society enforces them, but their final result is to place socially imposed costs on the individual that can change the potential benefit of selfishness. It follows that, even though agents could in principle have a positive utility for acting like self-seeking egotists which lie, cheat, steal, and freeload if they can, society will impose costs that will shift the rational decisions of agents to reach a socially more desirable outcome. In short, **opportunistic behaviour can be controlled through social rules enforced by all peers through social institutions**. However, this implies some measure of *accountability*, and this can be difficult to achieve in peer-to-peer settings (see Section A.3.4).

The issue of how social norms emerge from individual behaviour is directly analysed in [124], where [124] posits that social norms are created to "re-internalise" externalities. This happens usually when a given social action imposes externalities on many individuals in a group, but the rights to control the execution of the action cannot be easily established. Thus, the costs of establishing rights of control over the action make it unprofitable for any of the affected individuals to attempt to do it. And since nobody can claim rights of control over it, nobody can attempt to profit from doing so.

It follows that it is not enough for a social rule to be useful and necessary: it is necessary to for the social group to be able to overcome the freeloader problem endogenous to its enforcement. If a given member of society takes upon herself to punish social rule violators, the rest of the community can freeload on her and not participate in the punishment of violators[4].

A.2.6 Currency

Conventionally, currency in economies is used as a *medium of exchange*, as a *unit of account* and as a *store of value*.

[4]There is evidence, however, that punish non-cooperators even in one-shot interactions [109]. This *altruistic punishment*, that in part explains the high levels of cooperation that human communities can sustain, has been validated by anthropological models of cultural evolution [57].

- **Medium of exchange**: This means that, conventionally, money simplifies trading by permitting barter to evolve into a much more flexible transaction involving easily convertible units. This relaxes the need for the two parties of a bartering relationship to have a coincidence of wants before trading can take place, and thus enables widespread and efficient economic activity.

- **Unit of Account**: Roughly, this means that currency must be able to support pricing. Therefore, currency can be used to describe the value of products and services.

- **Store of Value**: To act as a store of value, a commodity, a form of money, or financial capital must be able to be reliably saved, stored, and retrieved - and be predictably useful when it is so retrieved.

Barter transactions can be broken up into atomic transactions mediated by currency, thus alleviating the need for finding a simultaneous coincidence of wants. This, in itself, can help increase market efficiency significantly. However, the use of currency allows time-deferred enjoyment of contributions made to the system. If currency can be used as a store of value, peers can choose to store it, so that it can be used when needed. Thus, peers can continuously participate in the peer-to-peer overlay, accumulating currency as they contribute resources to the system which they can enjoy at a later time. However, the use of market also brings with it problems that barter-based systems do not have. The destruction or counterfeiting of currency are usual examples of this.

Another currency-related source of inefficiency arising in virtual economies is that there might be *starvation*: It is possible for a peer to have available resources and another peer that needs them to be unable to use them because it has insufficient funds to do so. Additionally, if money can be used to store value, economic actors may attempt to *hoard* money in order to manipulate the market or influence the price of products or services. Then, economies where money has the ability to store value are susceptible to inflation/deflation cycles triggered by market resonances with the circulation money supply. A common way to address this shortcoming is the direct modification of the money supply, an act usually carried out by central banks).

A.2.6.1 Demurrage

A solution that has been suggested for the problems noted in Section A.2.6 is *demurrage* [134]. Demurrage is a cost associated with holding currency outside the market. It works as a negative-rate interest, leading to a reduction in the amount of currency proportional to the length of time it remains stored.

In a currency system with demurrage, capital is a much better store of value than currency, and such a system provides strong incentives for investment. In demurrage free systems, speculators can invest in high yield, short term enterprises, and sell their shares for money before they become unprofitable due to unsustainable practices. This money will not lose value, and thus real gains can be made supporting non-sustainable infrastructure. In demurrage-driven systems, however, any money earned by any investor should be immediately re-invested, as storing it will lead to lost value (demurrage makes hoarding unprofitable). Thus, demurrage forces economic actors to focus in long-term sustainable growth investments (property, infrastructure, education, technology, health, etc), rather than immediate gains that may be unsustainable. Demurrage theorists posit that demurrage acts like monetary inflation: stimulating the circulation of the currency, encouraging economic activity, and increasing employment. However, its effects have not been extensively studied, nor have its benefits (or disadvantages) been rigorously demonstrated.

Obviously, if currency loses value very quickly after it has been spent, it can fulfil the first two characteristics above while giving up the third one. It may advantageous to the design of currency based resource allocation systems to decouple these functions, so demurrage may be an interesting tool for incentive mechanism design. Practical research systems that include demurrage are detailed in Section 2.5.6.

A.2.7 Prices

Many resource allocation systems that make use of prices do so because they can take advantage of any one of the basic functions of prices:

- **Information diffusion**: Prices act as signalling devices, informing peers where to devote their local resources in order to increase system performance the most. Due to the interactions between supply and demand, scarce network resources will have increasing price, and thus will be more attractive for redistribution. On the other hand, idle resources would fetch low prices, providing an incentive for peers to use them. From a supplier's point of view, if resource prices are high (the

system is having resource shortage) it is in its best interest to free as much resources as possible to devote them to the system, to take advantage of the increased margins provided by high prices. If, however, prices are low, suppliers are provided an incentive to aggressively seek out as many consumers as possible, to maintain their income levels even if the revenue margin per unit traded is small. In this case, prices transmit information about user tastes (demand), network resource quality and availability (supply), and content and fragment availability (market state). In principle, prices can be overall indicators of a very wide range of system information.

- **Value estimation**: Through distributed price-setting through supply and demand, each peer's assessment of the current system state and the actions of other peers is integrated in a single indicator. Therefore, if information is distributed through the network in a quick and reliable fashion, prices will converge to the true value of resources: peers demanding prices too far away from the equilibrium prices will have a lower aggregate income, and correct accordingly. Therefore, the average margin that every peer can extract out of every transaction can only be realistically increased by reducing its costs. Market dynamics provide an incentive for peers to adopt the least costly methods for stream distribution, and to use available network resources for the most highly valued uses. This, in turn, promotes the most efficient utilisation of available peer bandwidth, and allows peers to benefit from the distribution of fragments to peers to which high quality transmission is possible, exploiting locality properties of the Internet. In the same token, market dynamics discourage the traversal of low delay fragments through high delay paths, as this destroys the effort (QoS) put by all prior peers involved in their dissemination, decreases their value, and consequently has an impact on their price.

- **Resource allocation**: Prices determine who gets what, and in which quantity. Thus, prices are a resource allocation tool that defines the distribution of income and resources. Higher contributions to the system made where and when most needed would fetch higher prices, directing resources where and when most needed. In essence, the property of prices of redistributing income is the fundamental reason for which they can be used as incentives. If the benefits that a peer obtains from the system do not depend in any way on its actions (that is, if prices are not used to redistribute income), then there is no reason for it to take the information carried by prices into account. The incentive for each peer to act in accordance with that information disappears, and prices are unable to fulfil any of its other functions. On the other hand, if the "income" of a peer depends on the difference between the prices that he receives for contributing resources to other peers and the prices it has to pay for consuming resources itself, then it has a strong incentive to attain the highest QoS for other peers (and itself, of course) while investing as little network resources itself. This involves increasing the supply of scarce resources, and in the process, helping the system deliver increased QoS in the case of increasing system load.

A.3 Social Rules and Institutions

After analysing market transaction governance, we now move to the analysis of governance through social institutions.

A.3.1 Social Capital and Trust

In its most basic definition, *trust* is a **relationship of reliance** between two parties that is assumed to extend into the future, based on what one party knows of the other. As such, it can be very helpful in controlling uncertainty, and consequently, transaction costs. Social interactions that create trust have *value*, in the sense that they affect the economic decisions of agents. Social organisation and social rules constrain and give context to the actions of economic agents, and are therefore important in the definition of the economy [86].

The value represented by social networks can be objectively analysed and quantified. It has usually been called *social capital*, and it has been repeatedly studied by researchers [143, 127, 126, 85, 124, 86]. By starting from rational decision-making, [86] proposes that social capital is defined by social structure and the "strength" of social links, and it mediates and facilitates social activities for the social actors involved. Just like financial capital (and other kinds of capital), social capital is *productive*: it allows the actors to achieve that in its absence would not be possible.

Social capital therefore represents the transformative capability of social networks. Just as physical capital is produced by the transformation of raw material into implements that can further production (tools and machines, for instance) and human capital is the transformation of humans into active actors

in the production process through skills and knowledge, social capital leverages the interconnectedness and trust relations of social structure in order to allocate production resources to social actors and bring about change, usually by helping disseminating information and arbitrating and facilitating resource access.

Another perspective on social capital comes from [56], where Bourdieu defines social capital as "the aggregate of the actual or potential resources which are linked to possession of a durable network of more or less institutionalised relationships of mutual acquaintance and recognition". It is clear from the previous arguments that, although in engineering and mathematical modelling terms it is tempting to frame social capital as a function of topology and interconnectedness alone, it is *trust* what transforms social relations into a transformative resource. In other words, the trust relations among social actors are fundamental on transforming a given social network in a source of social capital. On [257], Putnam focuses on this relationship by defining social capital as a set of features of social life that enable participants to act together more effectively and to pursue shared objectives. In general, Putnam identifies social capital as referring to social network connections and their attendant norms and trust.

Although Putnam recognises that social and monetary capital are certainly different, they can be made equivalent through a generalised barter structure. In particular, if a given social agent A does something for B and trusts B to reciprocate in the future, this establishes an expectation in A and an obligation on the part of B. This obligation can be conceived as a credit slip held by A for performance by B. If A holds a large number of these credit slips, for a number of persons with whom A has relations, then the analogy to financial capital is direct. These credit slips constitute a large body of credit that A can call in if necessary - unless, of course, the placement of trust has been unwise, and these are bad debts that will not be repaid [86]. Thus, in [86] Coleman identifies the equivalence of social and market capital as stemming from a generalised barter system based on *reciprocal altruism* (see Section A.4.3).

The social capital of reciprocal altruism in a given society is a function of two related but independent variables: the *trustworthiness* of the social actors (upon which rests the probability that contracted obligations will be repaid), and the actual extent of obligations held (the number of contracts each social actor has). Therefore, *trust* is a fundamental enabler of social capital, and will have a central role on any peer-to-peer system.

Another form of social capital is the *information flow* inherent in social construction, since an agent can use the same exchanges used to form social ties to obtain information that is individually interesting to it. Peers can use actual fragment exchanges in which they participate to ascertain supply and demand, and drive downloading decisions on these - this is the basis for the *rarest first* policy of BitTorrent. Additionally, the ability that a peer has to piggyback requests and replies for information in the basic network keepalive messages is a form of social capital.

Social norms, when effective, constitute powerful sources of support for social capital. As has been explained before (See Section A.2.1), norms can be used to control transaction costs through the reduction of externalities, and can even be used to build alternative allocation structures for cases where market governance is unsuitable.

A.3.2 Network Topology and Social Enforcement

Social structures (network topologies, in case of peer-to-peer networks) that exhibit **closure**[5] facilitate the formation of social capital [86]. Network closure is a necessary precondition for the stabilisation of social norms because it allows the actions of a given agent to have an effect on the utility of any other agent, even if this effect is indirect. This helps control externalities, one of the main problems for resource allocation on a market basis. If a peer, in contravention of a known social rule, exploits another peer and there are not enough influence paths through the network so that the wronged peer can retaliate against it, it might be difficult to make cheating unprofitable: the retribution cost that the cheating node suffers will be smaller than the benefit it gained from the exploitation. Therefore, network topologies without closure cannot support social norms, as they will be essentially unenforceable, and strategic peers will simply choose to disregard them.

Of course, if a sizable set peers fail to observe the social rule, there will be clear system-wide economic losses. However, this damage will be exerted not only on the exploiting nodes, but on all the

[5]**Network Closure** is a sociological concept that relates to a topological property of a directed graph. Basically, a social network has closure if the providers of a given peer p can be compelled to retaliate against it if p exploits any peer for whom it is a provider. From a graph theoretical point of view, a trust graph exhibits closure if it is possible to find directed paths (ideally, a large number of these) from every node in the outgoing node neighbourhood of p to every node in its incoming node neighbourhood (see Section 3.1.1). Thus, any peer v exploited by p can threaten to retaliate on one of its successors and this influence can propagate all the way back to p, multiplied by the number of available paths in the network.

rest of the peers that do follow the social rule. Thus, exploitative peers externalise their economic losses, and social norms cannot act: there is no way to punish noncompliance. Thus, in a society pervaded with externalities (or a topology not exhibiting closure), norm-violating behaviours can continue unabated because the victims can only punish the perpetrator personally, instead of leveraging their social network resources to retaliate upon the perpetrator[6].

Given that social capital depends critically on the interconnectedness of social networks, it is in many respects similar to a *public good*, and can fall victim to freeloading (underinvestment and exploitation, as detailed in the *Tragedy of the Commons*[7]). This is because the social structures that make possible social norms, and the social rules and sanctions that enforce them, do not benefit primarily the peers whose efforts actually build such structures, but rather, they benefit all peers roughly equally. For instance, in a Gnutella-like peer-to-peer system, if dense subnetworks between large numbers of nodes exist, these are the result of a small number of peers acting as *superpeers* and processing a disproportionally large number of queries. However, every single peer benefits from this. On the other hand, the benefit that a superpeer can extract of its own privileged position is limited, and only a subset of the benefits of the social capital crystallised in the complete network topology. Therefore, it may be perfectly rational for a superpeer node to relinquish this role and transform into a leaf node: its perceived loss in searching capabilities can be partially subsumed by the superpeers that remain in their roles, and, in its new role as a leaf node, it will enjoy greater available bandwidth.

In general, this is the case with social capital:

> *When an individual asks a favour from another, incurring in an obligation, he does so because it brings him a needed benefit; he does not consider that it does the other a benefit as well by adding to a drawing fund of social capital available in a time of need. If the first individual can satisfy his need through self-sufficiency, or through aid from some official source without incurring an obligation, he will do so - and thus fail to add to the social capital outstanding in the community. Similar statements can be made with respect to trustworthiness as social capital. An actor choosing to keep trust or not (or choosing whether to devote resources to an attempt to keep trust) is doing so on the basis of costs and benefits he himself will experience. That his trustworthiness will facilitate others' actions or that his lack of trustworthiness will inhibit others' actions does not enter into his decision. A similar but more qualified statement can be made for information as a form of social capital. An individual who serves as a source of information for another because he is well informed ordinarily acquires that information for his own benefit, not for the others who make use of him.*
>
> (quote from [86])

The fact that most social capital behaves as if it were a public good implies that it is fundamentally different from other kinds of capital with respect to incentive mechanisms. Social capital (such as gossip-based searching or information diffusion) is an important resource for peers, and may affect greatly their ability to interact with other peers and their perceived quality of experience. However, even though it is peers themselves who bring social capital into being through their interconnection topology and its related trust network, the benefits of network construction and maintenance (the actions that bring social capital into being) are largely experienced by peers other than themselves. Therefore, it is often not in their interest to bring the network into being in the first place: there is always an open incentive to freeload. In practice, most forms of social capital are created or destroyed as by-products of other activities, such as file transfer. This is the case of BitTorrent, for instance, where the swarm is set up for file transfer and only has meaning in terms of it: BitTorrent does not support the creation of social capital independently of file transfer (however, BitTorrent communities achieve this outside the confines of the peer-to-peer network).

There is, however, a private capital angle to social capital. In [127], Fukuyama discusses this in terms of selfish motivation: as every agent needs cooperation as a means of achieving its selfish ends, social capital (cooperation and trust) will be produced out of strictly self-seeking behaviour. In particular, Fukuyama makes the case that if repeated interaction is modelled using an iterated *Prisoner's Dilemma*, a *Tit-for-Tat* strategy would be conducive to a cooperative outcome. Thus, Fukuyama posits

[6] Of course, in social models this concept is important for models of legality, corruption and criminal behaviour.

[7] Hardin sates himself that he does not use "tragedy" as a synonym for sad or unfortunate. Rather, he uses it as an expression to identify a deterministic process that once set up, will inexorably continue. Hardin thus underlines the self-sustaining nature of the over-exploitation of shared resources.

that value systems and social constructs that support a reciprocative interaction policy would be favoured. In particular, Fukuyama proposes that value systems based on honesty, industriousness, and prudence can be explained as arising from reciprocity in market transactions. In a sense, Fukuyama posits that social norms and value systems evolved from reciprocity between individuals. For those humans that were born when they were already in place, these rules acted as control mechanisms that ensured the maintenance of the cooperative outcome. This means that one dimension of social norms is that they are self-sustaining strategies to control freeloading, and they can arise out of self-interest in situations where utility maximisation calls for a cooperative outcome.

Thus, **selfish behaviour of the peers can be used to control their own selfish behaviour**. The question arises then: How to develop overlay network protocols and peer behaviour algorithms so that, in the aggregate, freeloading is controlled? If social norms (in the shape of reputation systems and decision-making policies based on the reputation ratings of peers) are used to control freeloading, then it follows that the basic punishment that a peer can receive if it deviates from the node is disconnection from its trading parties: the loss of social capital.

A.3.3 Network Topology and Trust

The basic engineering appeal of social capital as a modelling construct is that it facilitates protocol design through the delegation of micromanagement to the peers themselves. This is done by explicitly modelling trust between agents, and through this, exercising distributed normative control. In this sense, our model shares with [127] the idea that social capital can help reduce the transaction costs associated with coordination mechanisms like contracts or marketplace interactions. Therefore, although markets can be used for resource allocation and distributed coordination, their costs can be reduced by an appropriate system of social rules. In this sense, social capital is useful because its open endedness: the cost of social rule violation is essentially unbounded, in the sense that it is very hard to predict accurately the amount of social capital loss that would be involved in the social retribution following the betrayal of a given economic agent. Like [127], we posit that social capital models manifest themselves as a "goodwill" of sorts that prevents the parties from taking advantage of unforeseen loopholes in market transactions.

In essence, **trust is the basic protocol element defining interaction between strategic peers**. According to [85, 124], trust has the following characteristics:

1. Placement of trust allows actions that would be otherwise impossible for boundedly rational agents. This is due to the fact that it relaxes the requirement for complete information on decision making scenarios, and therefore enables boundedly rational peers to conduct actions with only incomplete information.

2. If the *trustee* (the agent in whom trust is placed) is genuinely trustworthy, the *trustor* (the agent that takes a risk by placing trust in another agent) will be better off than if it had not trusted. Conversely, if the trustee is not trustworthy, the trustor will be worse off than if he or she had not trusted. However, if one of the agents is trustworthy and the other is not, the untrustworthy agent will be able to cheat the trustworthy, trusting peer and extract a high utility at a very low cost. In this sense, placing trust is strongly reminiscent of a *Prisoner's Dilemma*.

3. Trust, as an action, involves the voluntary placement of productive resources at the disposal of the trustee, with no hard commitment from it. Because of this, many trust situations involve a *Principal-Agent* situation with moral hazard (See Section A.4).

4. A time lag exists between the extension of trust and the result of the trusting behaviour. Thus, trust is a characteristic of time-extended markets, as opposed to spot markets.

If we view trust as an essential component for boundedly rational agents to develop effective and dependable interaction heuristics, a more in-depth analysis of trust and its relation to social capital may be useful for protocol design. In sociological terms, usually two kinds of trust are considered [257]:

1. **Thick** trust: trust between social agents that *know* each other, that is, that have interacted in the past and have a definite expectation on future interactions on the basis of this past history. Consequently, thick trust is a property of intimate social networks, and tends coincide with dense, cliqued network topologies.

2. **Thin** trust: trust between social agents that *do not know* each other. This kind of trust is just a baseline on the generalised trust that random community members have with one another, and tends to coincide with sparse, unclustered network topologies.

This notion can be further refined by explicitly adding a topological dimension. In [126], Fukuyama generalises the notion of "kinds of trust" by referring to *trust radii*. By this he means that a high-trust subnetwork of people, among whom co-operative norms operate, can be embedded in a generic social network with much lower mean trust[8].

Therefore, Fukuyama analyses social groups as circles of trust, where the groups are formed by selfish motivation, but externalities may be created that extend beyond the initial scope of the group members:

> *All groups embodying social capital have a certain radius of trust, that is, the circle of people among whom cooperative norms are operative. If a group's social capital produces positive externalities, the radius of trust can be larger than the group itself. It is also possible for the radius of trust to be smaller than the membership of the group, as in large organisations that foster cooperative norms only among the group's leadership or permanent staff.*
>
> (quote from [127])

Fukuyama states that the clustering defined by close-knit communities with ample social capital is independent from the social labelling that the community members themselves create, and to the organisational characteristics of the community. A large, geographically distributed society is then modelled as a series of trust circles with varying radii, that overlap each other through members that are part of distinct high-trust groups. These interlocking trust groups are brought together by transitive trust relations among the members that connect them.

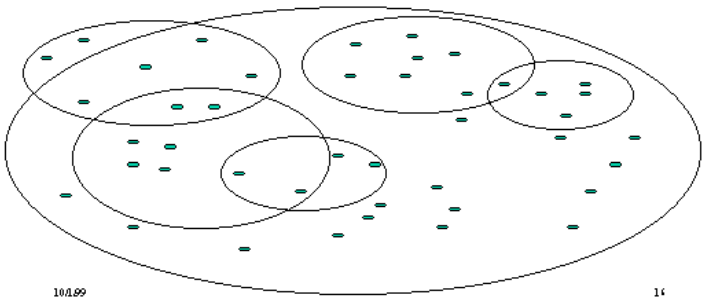


Figure A.1: The trust model of overlapping circles presented in [127], with trust radii dependent on the nature of the social relationship. (Image taken from [127])

The idea that different social links can be qualitatively different has led to the most common distinction established when discussing social capital from a network-topology standpoint: the difference between **bridging** and **bonding**. These different kinds of social links have the following characteristics:

1. **Bonding Links**: these are usually built with thick trust relations among members of homogeneous groups. They usually imply a strong sense of "group identity", and thus tend to favour cliquishness. A typical example of this in a human context would be ethnic fraternal organisations. Even though these links are highly reliable, peers may find that their usefulness is limited: as group members tend to be similar to one another and tend to develop in similar circumstances, what they can get from one another that they not already have is limited. Thus, these links are good for "getting by" [257]: they can be used as support for everyday requirements and activities, but are not useful for further development or access to new resources.

2. **Bridging Links**: these are usually built with thin trust relations between members of distinct groups. They require membership openness between groups, and tend to appear on socially diverse environments. In human societies, bridging social capital refers to relations with distant friends, associates and colleagues: links that are more functional or situational than ethnic or cultural. Even though these links may be unreliable, peers may find them extremely useful: as peers on each side

[8] Intriguingly, Fukuyama further suggests that in many Latin American societies, a small subnetwork diameter (a narrow radius of trust) tends to produce a two-tier moral system, with good behaviour reserved for family and personal friends, and a lower standard of behaviour in the public sphere. Thus, in the case of naturally evolved incentive systems, the interaction topology of the agents and the strategies defined by the incentive mechanism are interrelated and have a manifest feedback.

of the link tend to be different from one another and to develop in different circumstances, their opportunities for fruitful trade are very large. Therefore, these links may be weaker and more diverse, but are fundamental for "getting ahead" [257]: they can be used as gateways to new social worlds, to obtain new resources or information that would be otherwise out of reach. These correspond to the weak ties proposed in [143][9].

In terms of peer-to-peer system design, it is simple to envisage high-trust peer communities with low transaction costs, but pretty homogeneous in terms of their "QoS coordinates" and their content. These groups might be connected by long-distance relationships which might have lower trust levels, but that are critical for collapsing the entire peer-to-peer network into a *Small World* [44, 315]. Following [143], we have that trading within small-high trust cliques might be enough for peers to "get by", but finding new content quickly demands long-distance connections.

The opportunities for the application of sociological models of trust to peer-to-peer networks may be of great benefit, as they provide a natural framework for the expression of strategic peer behaviour while focusing on the equilibrium states of emergent behaviour. One abstraction that allows flexible algorithm design is trust and its attendant social capital. Even though not framed in the vocabulary of sociology, peer-to-peer researchers have approached trust issues repeatedly, and it remains a topic of ongoing research. For instance, in [316] Wilcox-O'Hearn identify strategic peer behaviour and its interplay with social capital as a public good as a fundamental problem and avenue for further research:

> *Perhaps the most challenging unsolved problem is that of mutual distrust. While a network architect is tempted to assume that all peers in the system behave as he designed them to behave, this assumption may prove fatal once a network is deployed into multiple disjoint administrative zones. A fundamentally related issue is that of "motivation to cooperate". Why does a peer choose to offer services to the network as well as to make requests of the network? Is there anything preventing a user from altering their copy of the software, or writing their own compatible implementation, which uses the resources of the other peers but refuses to provide its own resource to them? Also closely related is the notion of "attack resistance". If a peer can use the resources of other peers without offering them service in return, then it is able to act as a drain on the resources of the network as a whole, possibly constituting a denial-of-service attack on the entire network. On the other hand, if a peer can be coerced into cooperating, perhaps by cutting that peer off from the services of the network in retaliation for its lack of cooperation, how can we be sure that the same mechanism cannot be used to attack specific (innocent) peers, or even to attack the network itself?*
>
> (quote from [316])

This set of challenges has been successfully, but only partially, addressed by the latest peer-to-peer systems, like BitTorrent. There still remain, however, significant opportunities for research in this area.

A.3.4 Accountability and Identity

If one considers a peer-to-peer system as a distributed, engineered system, its main role is to perform efficient, predictable resource allocation with strategic (and/or untrustworthy) peers. Resource allocation has been traditionally approached through the notion of **accountability**: peers are made responsible for their resource usage. Typically, for centralised systems, this is done by separating the four basic components of an accountability system[10]:

- **Authentication**: Typical resource allocation systems rely on users having stable identities, and adequate credential management systems to manage this identities. Authentication is the process of ascertaining whether a user claiming certain identity is actually the user whose that identity belongs to, and it is performed by verifying that the credentials presented for a claimed identity match those credentials bound to that identity on a trusted repository.

- **Authorisation**: Once that identity has been ascertained (through authentication), typical resource access systems match identities to capability and access profiles. Thus, the authenticated user is

[9] Granovetter posits that the degree of overlap of two individuals' friendship networks varies directly with the strength of their tie to one another, and therefore it is these weak ties that have the greatest network-wide cohesive power.

[10] Usual AAA protocols include RADIUS (RFC 2865, RFC 2866), DIAMETER (RFC 3588), TACACS (RFC 1492, RFC 0927) and Cisco Systems'[6] TACACS+.

granted access to system resources depending on its identity, the current system state and a set of centrally-defined policies.

- **Accounting**: Once a user has been authorised to use system resources according to system-wide policy, its actual system resource consumption is tracked and recorded, usually for planning and billing purposes.

- **Auditing**: All system usage must be correlated with an authenticated user, its authorised resources and its actual resource usage. Because of this, resource allocation systems usually have auditing capabilities that can be used for fraud detection, usage anomaly detection and forensics.

These definitions, requiring central trust anchoring and repositories, are unwieldy for peer-to-peer systems. However, by distributing these centralised functions, peer-to-peer equivalents arise.

In a peer-to-peer system, for instance, there is no trust anchor for identity management. Usually, identity in peer-to-peer systems cannot be absolutely assured, and the system must settle for *pseudonymity* instead. However, the question still remains of how to bind pseudonyms to peers. This is usually done using asymmetric cryptography techniques (such as elliptic curve cryptography [226], RSA [267] or ElGamal [129]). In this case, although each peer can prove its own identity by using self-signed digital certificates, these can be changed at will and do not constitute protection against sybil attacks. However, we can leverage the concepts in Section A.3.1 for identity management. If a peer defines its own identity and then embeds itself in a social network, its relations with other peers will constitute social capital. If the peer were to create a new identity from scratch, this identity will not have the trust links that the previous one had - the social capital would be lost. Thus, the user behind the peer *loses* capital by abandoning an old identity and creating a new one. This social capital loss becomes a disincentive for short-lived identities, and becomes a self-imposed identity management system: it is in the best interest of nodes to maintain their own identities, at least for non-exploitative peers. Additionally, it is in the best interest of each peer within a reputation system to correctly identify which peers it is interacting with. Therefore, identity management (as detailed above) coupled with peer self-interest and risk minimisation constitute a distributed authentication system for strategic peers.

In the case of exploitative peers, two basic preconditions for this system to work would be:

1. **Social capital is always non-negative**.

2. **Unknown peers have zero starting social capital**.

Thus, if new peers do not have a "free" allocation of resources when joining the system, sybil attacks (*pseudospoofing*, in the vocabulary of Detweiler [1]) becomes unprofitable, and therefore burdened with a heavy negative incentive.

The peer-to-peer equivalent of authorisation is a reputation system working alongside a (possibly market-based) resource allocation system. Thus, the access that peers have to system resources is decided, on a peer to peer basis, by their previous actions. This implies that, on the basis of past experience, peers might refuse service, grant priority service or provide only best-effort service. Of course, a fundamental consideration at this point is *risk*: peers participating on a transaction could limit access to their resources to an amount roughly equivalent to their expected gain from it.

Once a peer has decided to grant access to its resources to another peer, it needs to be able to adequately measure how much resources were actually delivered. This might be much more complex than it might initially seem, as contention will make the provision of a given level of service much harder in some instances than in others. Therefore, a dynamic way to ascertain peer resource value is needed, and an immediate candidate is a market-based pricing system.

From the discussion above, we see that *identity* is defined by social embeddedness, trust and social capital, and can be *authenticated* in a peer-to-peer basis by using self-signed certificates. Resource access (*authorisation*) can be managed through a reputation system, and measured (*accounting*) through a currency-based exchange system or market.

It has been argued that market-based currency systems can be used for risk management and resource access as well [100]. However, to do so requires each transaction to build its own trust profile by starting with small value exchanges, incrementally increasing the exchanges as trust is accumulated. This has the advantage of requiring no persistent reputation storage, but may be slow to converge [11].

[11] One can find parallels between this mechanism and simple contract signing protocols in cryptography [278].

A.4 Reciprocity

As noted in Sections A.3 and A.2, reciprocity has been fundamental in the construction of human markets and societies. We now analyse the issue further.

A.4.1 Reciprocity, Reputation and Trust

Reputation Systems have been repeatedly proposed (and used) as a tool for facilitating cooperation and combating freeloading. In [233], Mui et al. propose a computational model for reputation that cleanly separates *reputation*, *trust* and *reciprocity*. The following definitions are used:

- **Reciprocity**: mutual exchange of deeds (such as favour or revenge).
- **Reputation**: perception that an agent creates through past actions about its intentions and norms.
- **Trust**: a subjective expectation an agent has about the future behaviour of another agent, based on the history of their encounters.

The first objective of reputation systems is to allow independent agents to conduct fruitful exchange transactions on environments with *moral hazard*[92] and *information asymmetries*[29].

In peer-to-peer networks, moral hazard presents when two peers come into an agreement, but each one stands to gain from ignoring it if the other one does not. This behaviour is commonly modelled with the *Prisoner's Dilemma* [179], and reputation mechanisms can deter it by acting as *sanctioning devices* [97]. If badly behaved peers are punished by other peers, and if the expected gains from cheating are smaller that the expected losses by punishment, the threat of public disclosure of a peer's cheating conduct provides sufficient incentive to cooperate.

Information asymmetry manifests itself naturally in the context of peer-to-peer systems, as usually there is information known to some, but not all, of the parties that stand to lose or gain as a result of it. Reputation mechanisms alleviate this problem by acting as *signalling devices* that induce peers to learn this hidden information in an indirect fashion.

If resource contribution on a peer-to-peer network is framed as an iterated *Prisoner's Dilemma* [39], the expectation of reciprocity for future interactions (including retaliation) can be strong incentive for cooperation [39]. However, the high churn rates typical of peer-to-peer networks mean that any given client may be interacting essentially with strangers [233][12]. The problem is further complicated if the subset of peers that have the content that a peer seeks are not the ones that it had contributed to in the past, thus making past contributions useless in the present. In these circumstances, a reciprocity system that allows contributions given to a peer to be taken into account by other peers may be the only way to foster cooperation between strategic peers seeking private utility maximisation.

Therefore, in general, reciprocity based schemes rely on peers being able to learn about the past behaviour of other peers and use this information in their interactions with them. We can distinguish two different kinds of reciprocity:

- **Direct** or **Private Reciprocity**, where the interaction between two peers is only influenced by the interactions between them in the past.
- **Indirect** or **Public Reciprocity**, where the interaction between two peers is not only influenced by the interactions between them, but by their interactions with all other peers as well.

Thus, direct-reciprocity schemes are more appropriate for long-lived relationships where there is ample opportunity for each of the peers to appropriately reciprocate the behaviour of other peers (see Section A.4.3 for a discussion on reciprocal altruism and the conditions on which it develops). However, it has been shown that direct reciprocity schemes for peer-to-peer networks tend to be slow (or may even fail) to converge[198].

A number of indirect reciprocity schemes have been proposed, differing mainly in the way that the reputation scores are calculated and propagated. Indirect reciprocity schemes, compared to direct reciprocity schemes, have much better convergence properties, but are susceptible to manipulation by slander, collusion or sybil attacks.

[12] Even though it has been shown that a sizable proportion of peers are highly stable [290], Stutzbach and Rejaie focus on information dissemination and bootstrapping. Relying exclusively on such long-lived peers for resource contribution would be to approach a client-server paradigm.

A.4.2 Direct Reciprocity

Empirical research has shown that human beings do not behave as self-interested, rational agents in the mathematical sense [285]. Rather, they exhibit non-negligible amounts of cooperation, even in the absence of a reputation system[55]. This might have its origin in natural selection [299], and, in the same spirit, is further reinforced by *memes* [96] that themselves have successfully replicated and thus have been favoured by cultural evolution. The *warm-glow* model [34] is based on this, and it postulates that socialised humans derive a direct benefit (a *warm-glow*) in the act of giving itself. Additionally, social embeddedness provides additional motivators for altruistic behaviour, in the shape of morals and value judgements which trigger emotions - in particular pride and shame [107].

The development of a theory of cooperation based on selfish individual drives (utility maximisation) mediated by a set of social restrictions (social capital or reputation) has lead to a large amount of research on reciprocal altruism. We now discuss some of this research.

A.4.3 Reciprocal Altruism

In [299], Trivers examines the emergence of **reciprocal altruism** in biological systems. Altruistic behaviour can be described as a set of actions performed by an organism that benefit another organism which does not carry the genes of the first. Therefore, in benefiting the second organism, the first one may incur opportunity costs or risks that may be apparently detrimental to it. This immediate analysis is, however, short sighted. Expected agent utilities will depend on the specific costs involved in each of the one shot games. In particular, Trivers notes that:

> *One human being saving another, who is not closely related and is about to drown, is an instance of altruism. Assume that the chance of the drowning man dying is one-half if no one leaps in to save him, but that the chance that his potential rescuer will drown if he leaps in to save him is much smaller, say, one in twenty. Assume that the drowning man always drowns when his rescuer does and that he is always saved when the rescuer survives the rescue attempt. Also assume that the energy costs involved in rescuing are trivial compared to the survival probabilities. Were this an isolated event, it is clear that the rescuer should not bother to save the drowning man. But if the drowning man reciprocates at some future time, and if the survival chances are then exactly reversed, it will have been to the benefit of each participant to have risked his life for the other. Each participant will have traded a one-half chance of dying for about a one-tenth chance.*
>
> (quote from [299])

Thus, altruistic behaviour increases the survival probability of organisms, and their general fitness. If there is a large probability for every member of the population of drowning, those individuals that risk their lives to save each other and reciprocate in kind will have greater probability of producing offspring than those that refuse to help others and face drowning on their own. However, the altruist decision is only warranted rationally if its benefit of the to the recipient is greater than its cost to the performer. Otherwise, even if the probability of ulterior reciprocation is close to 1, the net cost incurred by the altruist peer outweighs its benefit, and altruism does not take place.

Even if benefit of the to the recipient is much greater than the cost to the performer, altruism will only be rationally chosen if the probability of the recipient reciprocating is high enough. However, from the standpoint of the recipient, it might not be rational to reciprocate: it has already been saved, and therefore does not derive benefit endangering itself by reciprocating. There is an incentive for the recipient to cheat. In order for altruist behaviour to be sustained, cheating must have later adverse effects on the cheating recipient which outweigh the benefit of non-reciprocation. Then, selection will discriminate against cheaters if the social reprisal from the wronged individual (and possibly the social group) has later costs that exceed the benefit of failing to reciprocate. Thus, if individuals follow a *trigger strategy* where the altruist responds to the cheating by curtailing all future possible altruistic gestures to this individual, and the benefits of these lost altruistic acts outweigh the costs involved in reciprocating, the cheater will be selected against. On the other hand, those individuals who follow a strategy conducive to cooperation, exchange many altruistic acts and have increased chances of survival and reproduction - and cooperators will find their genes quickly disseminating[299].

In light of this, **reciprocal altruism can only exist in peer-to-peer systems with a reputation system and a trigger strategy**, even if it only relies in the private experience of each peer. If we define an **altruistic situation** as any one in which one peer can benefit a second one in a greater amount than the cost of the act to itself (again following [299]), altruism will be sustainable in the following circumstances:

- When the probability of encountering another peer in an altruistic situation is high
- When a given peer repeatedly interacts with the same small set of peers, in such a way that altruistic behaviour can be reciprocated
- When peers can recognise each other, and recall whether another peer is a cheater or not
- When pairs of peers play *symmetric* roles in altruistic situations, so that both are able to benefit each other roughly equally, at roughly equivalent costs.

In the biological world, this leads to a model for reciprocally altruistic behaviour based on a set of population-dynamic parameters. In the context of peer-to-peer systems, we can modify those and propose them as engineering objectives to support reciprocal altruism:

1. **Peers should have long lifetimes.** If peers have long lifetimes in the peer-to-peer system, the chance that any two peers will encounter many altruistic situations is maximised. If peers only track the reputations of peers they have directly interacted with, long lifetimes increase the probability of two peers meeting again after a previous altruistic episode, and thus being able to reciprocate. If peers track reputations of peers they have not directly interacted with, long peer lifetimes allow to check with experience the reputation ratings of peers, helping in the control untrustworthiness in reputation reporting.
2. **Peers should aggregate in communities.** If peers belong to the same community during all or a significant portion of their lifetime, the chance that any individual will interact repeatedly with the same set of neighbours increases. This allows a much better computation of reputation, and it allows trust anchoring to be done in a distributed fashion.
3. **Peers should be mutually dependent.** If a set A of peers can obtain from peers belonging to another set B significant amounts of resources, peers from A will tend to establish relationships with peers from B, and thus the rate at which they interact with each other will be larger than the interaction rate with nodes outside either A or B. Thus, peers from A and B will tend to interact more, and the chance they will encounter altruistic situations together will increase.
4. **Peers should be free not to cooperate with each other.** If some peers are obedient and other peers are strategic, a linear dominance [13], asymmetric hierarchy is formed: strategic peers can exploit obedient peers with no fear of retaliation[14]. Thus, strategic peers become dominant over obedient peers, but not vice-versa. In this circumstance, there is nothing the obedient peer can do to punish lack of reciprocation from a cheater peer, and this undermines the benefits of altruism.
5. **Peers should be free to cooperate with each other.** Reciprocal altruism relies on a peer being able to provide to another peer a service that it cannot perform itself. Ideally, if there is scope for cooperation between two peers, these two peers should be able to find each other, describe the scope of cooperation, and act upon it.

A.4.4 Indirect Reciprocity

Although direct reciprocity schemes have the advantage of being easily implementable on the peers, it has been shown that they have nontrivial convergence issues[198]. It follows that direct-reciprocity schemes are more appropriate for long-lived relationships where there is opportunity for each of the peers to appropriately reciprocate. For other situations, indirect reciprocity may be the only technique with adequate stability properties.

One of the most popular techniques for the implementation of indirect reciprocity in peer-to-peer networks is the use of *reputation systems*[15]. Reputation systems harness the bi-directional communication capabilities of the Internet to engineer large scale word-of-mouth networks [97]. Some of the most practically successful e-businesses are based upon reputation systems, such as eBay [264], and Amazon.

In [100], Dingledine et al. proposes the following list of requirements for a reputation system:

- **Peer History Awareness**: The reputation system should distinguish new entities of unknown quality from entities with bad long-term performance.

[13] In a linear dominance hierarchy, each individual has a rank in the hierarchy, so that higher-rank animals have privileges over lower-rank animals. A typical example of this is the *pecking order* amongst poultry.

[14] Of course, obedient peers are assumed to be "programmed" for cooperation. Otherwise, the system itself is unsustainable.

[15] The equivalent sociological/anthropological model for this is the *image scoring* model by Kandori [187].

- **Known Accuracy**: The reputation system should report, additionally to reputation scores, the confidence of any given score - the likelihood this score corresponds to reality.
- **High Precision**: The reputation system should be stable, returning the same ratings consistently for the same peers under the same system state.
- **Weighting towards current behaviour**: The system should reflect recent trends in entity performance quickly. Thus, entities that might have behaved well for a long time but start behaving maliciously are quickly recognised and their trust values modified accordingly.
- **Efficiency**: The system should not require system-wide synchronisation or signalling, complex distributed calculation or any kind of centrally coordinated computation. It must only rely on local peer knowledge, and should reconverge quickly after state changes.
- **Robustness against lies, slander and collusion**: The system should resist any attempts by a group of spoofed peers under common control (sybil attack), or a set of colluding peers to influence reputation ratings in any way. Reputation ratings should truthfully reflect the actual quality of the peers.
- **Anomaly detection capability**: The system should be able to detect outlying data points representative of anomalous behaviour and act accordingly.
- **Truthfulness**: The reputation system should ensure that peers that report the performance of other peers do so in a completely free manner: no peer should be punished for truthfully revealing system information. This might involve making it impossible for a rated peer to learn the ratings that each of the rating peers gave[16], or it might involve making it unprofitable for peers to retaliate against other peers who tell the truth.
- **Smoothness**: Adding a single rating or a small number of ratings should not severely distort the state of the reputation system.
- **Understandability**: Any reputation system traces its rationale back to the incentives of the users behind the peers. Thus, it should be relatively easy to explain to the users what reputation scores mean, and the policies that peers follow as a result.
- **Verifiability**: The peers should be able to provide some sort of evidence to support their reputation management decisions, even though this evidence is not unconditionally trusted. This implies that transactions must leave behind verifiable evidence, and disputes can be resolved by weighting the evidence presented by the relevant parties.
- **Feedback Elicitation**: The reputation system should ensure that peers report back the results of their interaction, even if they were successful and therefore have no incentive for retaliation.

It is important to note that, even though it might be a starting point for mathematical modelling, reputation scores cannot be universal, scalar quantities describing the overall "quality" of a peer. This is because, in order to be useful in predicting the outcome of an specific interaction, reputation scores must pertain to that particular kind of interaction. Thus, a flexible reputation system that can be used to predict the outcome of different kinds of related interactions in different contexts will need to be, at least, a vector of different context-sensitive reputation scores. In particular, it is important that the reputation system is able to take into account the *credibility* of each peer when processing its reports. Architecturally, this means that reputation score and credibility need to be separate dimensions of the reputation system.

Additionally, it is possible that peers that behaved in a given way when the system was in a given state can behave in a completely different way when the system is in a different state. Then, if reputation system measurements are supplemented with system state information, their predictive power increases, and it, the usefulness of the system.

In their most general conception, reputation systems should cluster similar transactions carried out in similar system states to maximise their predictive capability. Usually, transactions are classified as belonging to given categories, and similar transactions are aggregated over these categories. However, it may be that the categories for clustering are unknown or unclear; and in that case, *Data Mining* can be used to extract those variables that hold greatest predictive power with respect to the transaction outcomes of interest in a given context.

[16] One way of achieving this would be the use of algorithms like Chaum's unconditional secrecy channel[68].

A.5 Social Enforcement and Reputation

As explained in Sections A.2.1 and A.3.1, market based systems usually manage their transaction costs by means of **social rules**. These rules essentially increase the capacity of boundedly rational agents to predict the outcome of a given transaction, and thus to minimise their exposure to risk. This reduces their policing and exchange costs.

In the case of streaming peer-to-peer systems, it is clear that, while market-inspired techniques are useful for the measurement of value and for resource allocation, they might insufficient for the establishment of adequate procedures to ensure system stability, particularly in the presence of arbitrarily hacked clients behaving in arbitrary ways. This situation can be vastly improved by using a reputation system.

Essentially, peers use the reputation system to minimise the impact of any attack on the system by minimising the impact of any attack on themselves, and thus actively managing their own risk.

A.5.1 Markets and Altruism

It is tempting to think that a pure altruism system supported by a reputation system may be sufficient to maintain the health of the system. In over-provisioned systems, or in systems where QoS is not delivered real time, this might be the case: an example in hand is the seeding of torrents in BitTorrent (see Section 2.6.1). However, in systems where scarcity (in this case, congestion and application layer contention) needs to be managed much more effectively, these simple systems tend to under-perform. This is not because reputation systems are inherently worse or more limited than market systems. The reason for this is much more fundamental: it is because, in transactions based on pure altruism, **it is usually very difficult to provide the relevant peers with the information that is relevant for the response to achieve any given engineering objective**. It is very difficult for individual peers to judge what is globally desirable, or what local actions can be taken that will benefit the system the most. Boundedly rational peers are necessarily limited; usually these peers cannot envisage or control the more distant effects of their actions. This might only imply a small performance hit when considering non-real time systems, but for real time overlays it is a fundamental engineering objective. For instance, in BitTorrent, the topological position of a seeder is of no consequence, as long as it can send missing rare fragments every now and then. In real time streaming systems, it is of paramount importance: bandwidth rich peers can become de-facto local distribution centres whose content will be defined by the community they are serving. It does not matter that an appropriate number of real time streaming peers "seed" a particular stream, if these peers are not positioned in the network in such a way that their combined effort can be amassed to generate a high-QoS stream for the greatest number of consumers. This is a global property, that depends not only in the local decisions of the peers, but on the traffic conditions in a given network, a given AS or even the Internet at large, and the demand patterns of the users. Thus, even if peers are assumed to be perfectly altruistic, a market-based incentive mechanism system is useful to help them find high-QoS and high-reliability paths at the overlay level.

This is because incentives transmitted through the price system are automatically accompanied by the information that is relevant to the effective operation of the incentive. Thus, high-QoS paths represent better cost-benefit trade routes, and the consuming end peer will prefer them over others. Paths spanning trustworthy peers will be preferred over intermittent, unreliable paths, just because they will have a better cost-benefit ratio. However, suboptimal paths will not be ignored: they will used as well, as long as the increased cost is balanced by increased redundancy or independence from local Internet effects, such as congestion hot-spots or misconfigured routers/firewalls.

B
Simulation Parameters

B.1 Simulation Parameters for Section 7.2.3

$$\begin{bmatrix} \beta_1 \\ \beta_2 \\ \beta_3 \end{bmatrix} = \begin{bmatrix} 12.519 \\ 12.426 \\ 20.946 \end{bmatrix}$$

$$\begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \end{bmatrix} = \begin{bmatrix} 0.55843 \\ 0.73457 \\ 0.71524 \end{bmatrix}$$

$$\begin{bmatrix} \delta_1 \\ \delta_2 \\ \delta_3 \end{bmatrix} = \begin{bmatrix} 0.70421 \\ 0.75578 \\ 0.50354 \end{bmatrix}$$

$$\begin{bmatrix} \gamma_1 \\ \gamma_2 \\ \gamma_3 \end{bmatrix} = \begin{bmatrix} .53454 \\ .20000 \\ .46675 \end{bmatrix}$$

$$\begin{bmatrix} q_{111} & q_{112} & q_{113} \\ q_{121} & q_{122} & q_{123} \\ q_{131} & q_{132} & q_{133} \end{bmatrix} = \begin{bmatrix} 0.00000 & 1.00000 & 1.67559 \\ 1.53490 & 0.00000 & 1.32990 \\ 1.87273 & 1.34256 & 0.00000 \end{bmatrix}$$

$$\begin{bmatrix} q_{211} & q_{212} & q_{213} \\ q_{221} & q_{222} & q_{223} \\ q_{231} & q_{232} & q_{233} \end{bmatrix} = \begin{bmatrix} 0.00000 & 2.00000 & 2.35860 \\ 2.62494 & 0.00000 & 2.53543 \\ 2.34937 & 3.65700 & 0.00000 \end{bmatrix}$$

$$\begin{bmatrix} p_{111} & p_{112} & p_{113} \\ p_{121} & p_{122} & p_{123} \\ p_{131} & p_{132} & p_{133} \end{bmatrix} = \begin{bmatrix} 0.077206 & 0.096902 & 0.064249 \\ 0.097260 & 0.081284 & 0.098252 \\ 0.066885 & 0.090100 & 0.068617 \end{bmatrix}$$

$$\begin{bmatrix} p_{211} & p_{212} & p_{213} \\ p_{221} & p_{222} & p_{223} \\ p_{231} & p_{232} & p_{233} \end{bmatrix} = \begin{bmatrix} 0.11022 & 0.11602 & 0.18819 \\ 0.13559 & 0.12398 & 0.13065 \\ 0.16901 & 0.18183 & 0.14284 \end{bmatrix}$$

Nomenclature

α_i Per-site diminishing traffic volume benefit for site i (Managed Overlay Model), page 128

β_i Total proportional traffic benefit for site i (Managed Overlay Model), page 128

χ An isomorphism for graph $\mathcal{G}$, page 55

$\Delta\mathsf{Q}_i(k)$ Inter-rank quality improvement function for rank k (Auction-based Swarming Model), page 98

δ_i Total diminishing traffic volume benefit for site i (Managed Overlay Model), page 128

$\delta_G^{\min}$ Minimal *Grim Trigger* discount parameter for cooperation (Hidden Action Model), page 124

$\delta_T^{\min}$ Minimal *Tit-for-Tat* discount parameter for cooperation (Hidden Action Model), page 125

η_0^k Magnitude parameter for flow k (Elasticity-based Model), page 136

η_k^k Price elasticity of demand of flow k (Elasticity-based Model), page 136

η_ξ^k Cross elasticity of demand of flow k with respect to flow ξ (Elasticity-based Model), page 136

$\frac{R}{F}$ Proportion of reciprocators vs. freeloaders (*PledgeRoute*), page 82

γ_i Per-site diminishing traffic quality benefit for site i (Managed Overlay Model), page 128

$\hat{\mathcal{B}}_i$ Total cost that the ESP would have had to pay to all ISPs for all the traffic terminating at site i had the budget condition not been binding (Managed Overlay Model), page 134

$\hat{\Omega}^i(q_c)$ q_c-quality level set for peer n_i in exact proportional quality mapping, page 96

$\hat{\Omega}^i(q_j)$ Feasible proportional quality subspace for winning bidder p_j in auctioneer n_i in exact proportional quality mapping, page 96

$\hat{b}_{ski}$ Amount of traffic flow from site k to site i over ISP s that would have been allocated had the budget condition not been binding (Managed Overlay Model), page 134

$\hat{s}(\mathsf{k})$ Aggregate valuation of a social outcome k to all agents, according to their reported valuation functions in an economic mechanism, page 65

$\hat{w}_{ij}$ The amount of contributions that n_j has given to n_i as response to reciprocative requests in a contribution network $\mathcal{G}$ (*PledgeRoute*), page 71

κ Basic amount of altruistic service (*PledgeRoute*), page 76

$\Lambda(\mathsf{S})$ Link neighbourhood of a connected subgraph S, page 52

$\Lambda(n_i)$ Link neighbourhood of node n_i, page 52

$\mathbb{P}_{ai}$ The power set of P_{ai}, page 57

$\mathcal{A}_j$ Set of possible actions for player p_j in a strategic game G, page 58

$\mathcal{B}_i$ Total cost that the ESP will have to pay to all ISPs for all the traffic terminating at site i (Managed Overlay Model), page 134

$\mathcal{C}$ Currency system, page 114

$\mathcal{D}_{ij}(w_{ji})$ Expected number of chunk slots that a request from bidder p_j will require for service in auctioneer n_i if it has previous contributions w_{ji} (Auction-based Chunk Swarming Example), page 103

$\mathcal{E}$ Set of all ESPs (Managed Overlay Model), page 128

$\mathcal{G}$ Directed Graph, page 51

$\mathcal{G}_i$ Partial view that peer n_i extracts of contribution network $\mathcal{G}$ (*PledgeRoute*), page 73

$\mathcal{H}$ Interest digraph: an edge (n_i, n_j) in $\mathcal{H}$ implies that peer n_i is interested in requesting services from peer n_j (*PledgeRoute*), page 82

$\mathcal{I}$ Set of all ISPs (Managed Overlay Model), page 128

$\mathcal{K}$ Set of all possible social outcomes in an economic mechanism, page 65

$\mathcal{L}_A$ Lagrangian for the client utility (Hidden Action Abstract Model), page 61

$\mathcal{L}_E$ Lagrangian for the constrained ESP model (Managed Overlay Model), page 133

$\mathcal{L}_H$ Lagrangian for the client utility (Hidden Action Model), page 116

$\mathcal{L}_I$ Lagrangian for the ISP profit optimisation model (ISP Profit Optimisation Model), page 142

$\mathcal{M}$ Market system, page 114

$\mathcal{O}$ Set of all possible outcomes of a strategic game G, page 58

$\mathcal{P}(A)$ The *power set* of A - the set that contains as elements all the subsets of A, page 57

$\mathcal{Q}$ Service Quality space in exact proportional quality mapping, page 96

$\mathcal{S}_j$ Set of possible strategies for player p_j in a strategic game G, page 58

$\mathcal{U}(n_i, P)$ Unvisited neighbour set of n_i for path P (*PledgeRoute*), page 73

$\mathcal{W}(P_{ij})$ Maximum contribution transfer capacity along path P_{ij} (*PledgeRoute*), page 79

$\mathcal{W}(s, i)$ Maximum contribution transfer capacity between a peer n_s and another peer n_i (*PledgeRoute*), page 78

$\mathcal{Z}$ Set of possible all type player type combinations in a strategic game G, page 58

$\mathcal{Z}_j$ Set of possible types for player p_j in a strategic game G, page 58

$\mathscr{N}$ Set of all routers in an ISP (Managed Overlay Model), page 128

$\mathscr{P}_{ai}$ The set of edge-disjoint paths between n_a and n_i that maximises flow reputation from n_a to n_i, page 57

$\mathsf{c}_j(s_{-j})$ Cost that can be applied to agent p_j as a function of the strategies of all peers except itself, in an economic mechanism, page 65

$\mathsf{D}(t)$ Transaction Delay cumulative distribution function (Hidden Action Model), page 119

$\mathsf{d}(t)$ Transaction Delay probability density function (Hidden Action Model), page 119

F_i Quality estimation function for peer n_i in exact proportional quality mapping, page 96

F_i^{-1} Inverse quality mapping correspondence for peer n_i in exact proportional quality mapping, page 96

$\mathsf{f}_a(\mathcal{G}, n_i)$ Anchor-flow reputation function for graph $\mathcal{G}$ anchored in n_a and evaluated at n_i, page 57

G A strategic game, page 58

h_j Part of a quasilinear utility function that represents a monetary payment, page 58

h_j Payment to agent p_j in an economic mechanism, page 65

k Social outcome in an economic mechanism, page 65

L Link set of a directed graph $\mathcal{G}$, page 51

L Set of all links in an ISP (Managed Overlay Model), page 128

$\mathsf{m}(n_i)$ Protocol performance measure (Auction-based Chunk Swarming Example), page 106

N Node set of a directed graph $\mathcal{G}$, page 51

N Set of all sites in the ESP (Managed Overlay Model), page 128

$\mathsf{o}(s)$ Outcome function that maps a strategy profile s to its outcome in a strategic game G, page 58

P Set of players in a strategic game G, page 58

p_j A player in a strategic game G, page 58

P_{ai} The set of all edge-disjoint paths between n_a and n_i on a graph $\mathcal{G}$, page 57

$\mathsf{Q}_i(k)$ Rank quality mapping function for rank k (Auction-based Swarming Model), page 98

$\mathsf{r}(x)$ Layer 3 RTT probability density function (Hidden Action Model), page 118

S A connected subgraph of a directed graph $\mathcal{G}$, page 52

s_1 Previous Debt Reciprocation Policy (*PledgeRoute*), page 82

s_2 Previous Contribution Reciprocation Policy (*PledgeRoute*), page 82

T_k^B Data point vector for bandwidth demands (Elasticity-based Model), page 138

T^p Data point matrix for prices (Elasticity-based Model), page 138

$\mathsf{V}_{-j}(\mathsf{k})$ Total value achieved by an economic mechanism if agent p_j does not participate, page 66

μ_{ij} Expected number of won bids (Auction-based Swarming Model), page 91

μ_{ij}^U Expected utility obtained from all won bids (Auction-based Swarming Model), page 94

ν Price that bidder p_j pays for participating in a single-item Vickrey auction, page 63

ω_j^* Resource allocation for winning bidder p_j in exact proportional quality mapping, page 96

Ω^i Resource space of peer n_i in exact proportional quality mapping, page 96

Ω_k^i Resource subspace k of peer n_i in exact proportional quality mapping, page 96

$\oplus$ Operator that aggregates paths to give end-to-end values (*PledgeRoute*), page 78

$\otimes$ Operator that aggregates links to give path values (*PledgeRoute*), page 78

ϕ Agent effort in the PA model, page 60

ϕ Level of effort by the server (Hidden Action Model), page 115

ϕ_ξ Per-unit infrastructure bandwidth cost for flow ξ for the ISP (Profit Maximisation Model), page 142

$\Pi(\mathsf{S})$ Node neighbourhood of a connected subgraph S, page 52

$\Pi(n_i)$ Node neighbourhood of node n_i, page 52

ψ Outcome-dependent payment in the PA model, page 60

ψ Payment given by the client to the server (Hidden Action Model), page 115

ρ_{stop} Probability that the trust path discovery random walk terminates at any given peer (*PledgeRoute*), page 73

σ_{ij}^2 Variance in the number of won bids (Auction-based Swarming Model), page 91

χ_{CTRM} Contribution compensation in case of CTRM transaction interruption (*PledgeRoute*), page 78

χ_{PAM} Contribution fine to act as a negative incentive to the termination of PAM walks (*PledgeRoute*), page 77

υ_{ij} Maximum amount of unreciprocated contributions that n_i will give to a neighbour n_j in a contribution network $\mathcal{G}$ (*PledgeRoute*), page 71

φ Winning bid "rank offset", $\varphi = N_i^j - M_i$ (Auction-based Swarming Model), page 90

ϑ_j Best response function for player p_j given its type and the strategy and type profile of all other peers in a strategic game G, page 59

ζ_j Type of a player p_j in a strategic game G, page 58

B_k Total amount of traffic flow over origin-destination pair k (Elasticity-based Model), page 135

b_k^m Traffic flow over origin-destination pair k corresponding to ESP m (Elasticity-based Model), page 135

B_{nm} Trail between nodes n_n and n_m in a directed graph $\mathcal{G}$, page 52

b_{ski} Amount of traffic flow from site k to site i over ISP s (Managed Overlay Model), page 128

$C(n_i)$ Stream Lag Stretch (Auction-based Chunk Swarming Example), page 106

c_l Unit ISP cost per unit of bandwidth transferred over infrastructure link l (Profit Maximisation Model), page 141

c_n Unit ISP cost per unit of bandwidth routed over infrastructure router n (Profit Maximisation Model), page 141

$CTRM$ Contribution Transfer Request Message (*PledgeRoute*), page 74

D Chunk delivery delay (Hidden Action Model), page 118

$d(n_j, n_k)$ Synthetic coordinate delay from n_j to n_k (Auction-based Chunk Swarming Example), page 102

$f(\mathsf{N})$ reputation function defined over N, the set of nodes of a graph $\mathcal{G}$, page 54

$f^{V^i}(v)$ Bid value distribution at auctioneer n_i (Auction-based Swarming Model), page 88

$F_{(k)}^{V^i}(v)$ Cumulative distribution function for the k-th order statistic of random variable V^i (Auction-based Swarming Model), page 90

$f_{(k)}^{V^i}(v)$ Probability density function for the k-th order statistic of random variable V^i (Auction-based Swarming Model), page 90

$f^{W^i}(v)$ Past contribution distribution at auctioneer n_i (Auction-based Swarming Model), page 98

F_i Transaction Fail message created by i and sent towards the CTRM originator s (*PledgeRoute*), page 74

$g(P_{ai})$ The *reputation flow* function of $\mathsf{f}_a(\mathcal{G}, n_i)$ ($g(P_{ai}) : \mathsf{P}_{ai} \mapsto \mathbb{R}$), page 57

H_t CTRM Header (*PledgeRoute*), page 75

$L_c(c_n, n_k)$ Continuous chunk lag for chunk c_n at peer n_k (Auction-based Chunk Swarming Example), page 102

$L_p(n_i)$ Discrete Peer Lag (Auction-based Chunk Swarming Example), page 106

$L_p(n_j, k)$ Continuous peer lag for peer n_j with a lag window of k chunks (Auction-based Chunk Swarming Example), page 102

$L_s(n_i)$ Stream Lag (Auction-based Chunk Swarming Example), page 106

M_i Contribution Transfer Request Message (CTRM) after being encapsulated by peer i (*PledgeRoute*), page 74

M_i Total number of bids that auctioneer n_i receives (Auction-based Swarming Model), page 88

M_s Contribution Transfer Request Message (CTRM) at its source, s (*PledgeRoute*), page 74

M_t Contribution Transfer Request Message (CTRM) response at its source, the destination peer t (*PledgeRoute*), page 74

m_{ji} Total number of bids that bidder p_j sends to auctioneer n_i (Auction-based Swarming Model), page 88

N_i^j Total number of competing bids that auctioneer n_i receives, as assessed by p_j (Auction-based Swarming Model), page 88

$N_{\mathcal{H}}$ Average peer outdegree in $\mathcal{H}$ (*PledgeRoute*), page 82

N_i Total number of bids that auctioneer n_i receives (Auction-based Swarming Model), page 88

N_s Number of peers amongst which the server upload is shared (Hidden Action Model), page 118

P Profit for the ISP (Profit Maximisation Model), page 141

p Probability of the client getting high quality service as a function of the effort by the server (Hidden Action Model), page 115

$p_j^{t+1}(s)$ Probability that a PAM starting at peer n_s visits peer n_j at time t (*PledgeRoute*), page 73

p_L Packet loss probability at layer 3 (Hidden Action Model), page 118

p_ξ Per-unit bandwidth price for flow ξ for the ISP (Profit Maximisation Model), page 141

P_{ij} Contribution path from n_i to n_j in a contribution network $\mathcal{G}$ (*PledgeRoute*), page 70

P_{nm} Simple path between from n_n to n_m in a directed graph $\mathcal{G}$, page 52

p_{ski} Cost per unit bandwidth that the ESP must pay to transfer data from k to i over ISP s (Managed Overlay Model), page 128

PAM Pledge Announcement Message (*PledgeRoute*), page 73

q Principal income in the PA model, page 60

q QoS enjoyed by the client (Hidden Action Model), page 115

q_T^i Target aggregate quality for peer n_i in exact proportional quality mapping, page 96

q_{ski} Benefit that a unit bandwidth traffic flow from site k to site i over ISP s provides to the ESP (Managed Overlay Model), page 128

R^L Link routing matrix (Profit Maximisation Model), page 141

$R_{l\xi}^L$ Link routing matrix element: 1 if origin-destination flow ξ goes through link l, 0 otherwise (Profit Maximisation Model), page 141

R^N Node routing matrix (Profit Maximisation Model), page 141

$R_{n\xi}^N$ Node routing matrix element: 1 if origin-destination flow ξ goes through node n, 0 otherwise (Profit Maximisation Model), page 141

r_{ij}^t Probability that a PAM is forwarded from node i to node j at time t (*PledgeRoute*), page 73

R_s Service request message that includes a trust granting ticket obtained through a trust shift operation. (*PledgeRoute*), page 74

s A particular strategy profile on a strategic game G, page 58

S_c Chunk Size (Hidden Action Model), page 118

$s_j(\mathsf{k})$ Valuation of a social outcome k that agent p_j reports to the mechanism designer in an economic mechanism, page 65

s_j^+ Best response for player p_j given its type and the strategy and type profile of all other peers in a strategic game G, page 59

S_p Packet size at layer 3 (Hidden Action Model), page 118

S_r Request message size (Hidden Action Model), page 118

T_c Client to server throughput (Hidden Action Model), page 118

t_P Request processing time at the server (Hidden Action Model), page 118

$t_Q(n_j, n_k)$ Time from the start of the service interval T that will elapse before n_j finishes its chunk transfer to n_k (Auction-based Chunk Swarming Example), page 102

T_s Server to client throughput (Hidden Action Model), page 118

T_t Trust granting ticket T_t for the request R_s (*PledgeRoute*), page 75

t_{old} Message expiry timeout value for replay protection (*PledgeRoute*), page 73

t_{T} Transaction expiry timeout value for the soft reservation protocol (*PledgeRoute*), page 75

t_{RTO} Retransmission timeout (Hidden Action Model), page 118

t_{RTT} Layer 3 RTT (Hidden Action Model), page 118

U_c Utility function for the client (Hidden Action Model), page 115

U_c principal utility in the PA model, page 60

U_c^d Defection Utility (Hidden Action Model), page 123

U_i Utility that site i obtains from participation in the system (Managed Overlay Model), page 128

U_j Utility for bidder p_j in a single-item Vickrey auction, page 63

U_j Utility for player p_j in a strategic game G, page 58

$U_j(\mathsf{k})$ Utility of agent p_j for social outcome k in an economic mechanism, page 65

U_r Rationality utility for the PA model, page 60

U_r Rationality utility for the server (Hidden Action Model), page 115

U_r Reservation utility for the server (Hidden Action Model), page 115

U_s Agent utility in the PA model, page 60

U_s Utility function for the server (Hidden Action Model), page 115

$v_j(\mathsf{k})$ Valuation of a social outcome k by agent p_j in an economic mechanism, page 65

v_j Valuation for bidder p_j in a single-item Vickrey auction, page 63

$w_+(n_i)$ Pledged resources (weighted indegree) of n_i in a graph $\mathcal{G}$, page 70

$w_-(n_i)$ Social capital (weighted outdegree) of n_i in a contribution network $\mathcal{G}$, page 70

w_l Provisioned bandwidth capacity for ESP traffic over infrastructure link l (Profit Maximisation Model), page 141

w_n Provisioned bandwidth routing capacity for ESP traffic over infrastructure router n(Profit Maximisation Model), page 141

w_{ij} Contributions from n_i to n_j in a contribution network $\mathcal{G}$ (*PledgeRoute*), page 70

W_{nm} Walk between nodes n_n and n_m in a directed graph $\mathcal{G}$, page 52

X^{ij} Number of winning bids on auctioneer n_i that were sent by bidder p_j (Auction-based Swarming Model), page 88

Bibliography

[1] The Snakes of Medusa and Cyberspace: Internet identity subversion (Lance Detweiler), November 1993. URL http://www.interesting-people.org/archives/interesting-people/199311/msg00054.html. 165

[2] Akamai, 2007. URL http://www.akamai.com. 20, 126

[3] Azureus, 2007. URL http://azureus.sourceforge.net. 31

[4] BitComet, 2007. URL http://www.bitcomet.com. 31

[5] The World Factbook (Central Intelligence Agency, CIA), 2007. URL https://www.cia.gov/library/publications/the-world-factbook/index.html. 38

[6] Cisco systems, 2007. URL http://www.cisco.com. 164

[7] eMule, 2007. URL http://www.emule-project.net. 14

[8] etree, 2007. URL http://bt.etree.org. 34

[9] Facebook, 2007. URL www.facebook.com. 39

[10] Freenet, 2007. URL http://freenetproject.org. 23

[11] W3C - HTTP, 2007. URL http://www.w3.org/Protocols/. 24

[12] LimeWire, 2007. URL http://www.limewire.com/. 23

[13] Mininova, 2007. URL http://www.mininova.org. 18

[14] μtorrent, 2007. URL http://www.utorrent.com/. 31

[15] The Pirate Bay, 2007. URL http://thepiratebay.org. 18, 37

[16] PlanetLab, 2007. URL http://www.planet-lab.org. 32, 44

[17] W3C - SOAP, 2007. URL http://www.w3.org/TR/soap. 24

[18] Wikipedia, 2007. URL http://en.wikipedia.org. 23, 155

[19] W3C - WSDL, 2007. URL http://www.w3.org/TR/wsdl. 24

[20] BitTorrent protocol specification, 2008. URL http://wiki.theory.org/BitTorrentSpecification. 18, 30

[21] Original BitTorrent protocol specification, 2008. URL http://bittorrent.org/beps/bep_0003.html. 18, 30

[22] Limelight, 2009. URL http://uk.limelightnetworks.com. 126

[23] Glasnost, 2009. URL http://broadband.mpi-sws.org/transparency/bttest.php. 49

[24] PPLive, 2009. URL http://www.pplive.com. 126

[25] Switzerland, 2009. URL http://www.eff.org/testyourisp/switzerland. 49

[26] Twitter, 2009. URL twitter.com. 39

[27] E. Adar and B. Huberman. Free riding on Gnutella. Technical report, Xerox PARC, August 2000. 3, 14, 15, 16, 68, 113

[28] V. Aggarwal, O. Akonjang, and A. Feldmann. Improving User and ISP Experience through ISP-aided P2P Locality. In *Proceedings of 11th IEEE Global Internet Symposium 2008 (GI '08)*, Washington, DC, USA, April 2008. IEEE Computer Society. 49

[29] G. A. Akerlof. The market for 'lemons': Quality uncertainty and the market mechanism. *The Quarterly Journal of Economics*, 84(3):488–500, August 1970. 166

[30] A. A. Alchian and S. Woodward. *The Firm Is Dead; Long Live The Firm:* a Review of Oliver E. Williamson's "The Economic Institutions of Capitalism". *Journal of Economic Literature*, 26(1): 65–79, March 1988. 155

[31] E. Altman, T. Boulogne, R. El-Azouzi, T. Jiménez, and L. Wynter. A survey on networking games in telecommunications. *Comput. Oper. Res.*, 33(2):286–311, 2006. 9, 48, 49

[32] K. G. Anagnostakis and M. B. Greenwald. Exchange-based incentive mechanisms for peer-to-peer file sharing. In *Proceedings of ICDCS'04*, pages 524–533, Washington, DC, USA, 2004. IEEE Computer Society. 44, 114

[33] N. Andrade, M. Mowbray, A. Lima, G. Wagner, and M. Ripeanu. Influences on cooperation in BitTorrent communities. In *Proceedings of P2PEcon '05*, pages 111–115, New York, NY, USA, 2005. ACM Press. 34, 45

[34] J. Andreoni. Giving with impure altruism: Applications to charity and ricardian equivalence. *Journal of Political Economy*, 97(6):1447–58, December 1989. 167

[35] P. Antoniadis, C. Courcoubetis, and R. Weber. An asymptotically optimal scheme for P2P file sharing. In *2nd Workshop on the Economics of Peer-to-Peer Systems*, Harvard University, 2004. 9, 26, 27

[36] M. Ardron and B. Lietaer. Complementary currency innovation: Self-guarantee in peer-to-peer currencies. *International Journal of Community Currency Research*, 10:1–7, 2004. 25, 68

[37] V. P. Author, K. Kumar, K. Tamilmani, V. Sambamurthy, and A. E. Mohr. *Chainsaw: Eliminating Trees from Overlay Multicast*, volume 3640 of *Lecture Notes in Computer Science (Peer-to-Peer Systems IV)*, pages 127–140. Springer Berlin / Heidelberg, 2005. 16

[38] R. Axelrod. On six advances in cooperation theory. *Analyse and Kritik (Special Edition on the Evolution of Cooperation)*, 22, 2000. 41

[39] R. Axelrod and W. D. Hamilton. The evolution of cooperation. *Science*, 211, March 1981. 41, 68, 123, 166

[40] N. Balakrishnan and C. R. Rao. *Order Statistics: Theory and Methods*, volume 16 of *Handbook of Statistics*. Elsevier, 1998. 90

[41] H. Ballani and P. Francis. Towards a global ip anycast service. In *Proc. SIGCOMM 2005*, August 2005. 20

[42] S. Banerjee, B. Bhattacharjee, and C. Kommareddy. Scalable application layer multicast. *SIGCOMM Comput. Commun. Rev. (SIGCOMM 2002)*, 32(4):205–217, 2002. ISSN 0146-4833. doi: http://doi.acm.org/10.1145/964725.633045. 16, 19

[43] R. Banner and A. Orda. Multipath routing algorithms for congestion minimization. *IEEE/ACM Trans. Netw.*, 15(2):413–424, 2007. 49

[44] A.-L. Barabasi. *Linked: The New Science of Networks*. Perseus Publishing, Cambridge, MA, April 2002. 164

[45] W. J. Baumol. Williamson's "The Economic Institutions of Capitalism". *The RAND Journal of Economics*, 17(2):279–286, 1986. 155

[46] J. C. R. Bennett and H. Zhang. Hierarchical packet fair queueing algorithms. In *SIGCOMM '96: Conference proceedings on Applications, technologies, architectures, and protocols for computer communications*, pages 143–156, New York, NY, USA, 1996. ACM Press. 15

[47] K. Bennett and C. Grothoff. GAP - Practical Anonymous Networking. In *Designing Privacy Enhancing Technologies*, pages 141–160. Springer-Verlag, 2003. 28, 45

[48] J. Bentham. *An Introduction to the Principles of Morals and Legislation*. Philosophical Classics. Dover Publications, 2007 (first published in 1789). 13

[49] A. R. Bharambe, C. Herley, and V. N. Padmanabhan. Some observations on BitTorrent performance. *SIGMETRICS Perform. Eval. Rev.*, 33(1):398–399, 2005. 30

[50] R. Bindal, P. Cao, W. Chan, J. Medved, G. Suwala, T. Bates, and A. Zhang. Improving traffic locality in BitTorrent via biased neighbor selection. In *ICDCS '06: Proceedings of the 26th IEEE International Conference on Distributed Computing Systems*, page 66, Washington, DC, USA, 2006. IEEE Computer Society. 17, 35, 36

[51] A. Blanc, Y.-K. Liu, and A. Vahdat. Designing incentives for peer-to-peer routing. In *INFOCOM 2005, 24th Annual Joint Conference of the IEEE Computer and Communications Societies*, pages 374–385, March 2005. 41, 42

[52] B. H. Bloom. Space/time trade-offs in hash coding with allowable errors. *Commun. ACM*, 13(7): 422–426, 1970. 39

[53] B. Bollobas. *Random Graphs*. Cambridge University Press, 2001. 79

[54] J.-C. Bolot. End-to-end packet delay and loss behavior in the internet. In *SIGCOMM '93: Conference proceedings on Communications architectures, protocols and applications*, pages 289–298, New York, NY, USA, 1993. ACM. 118

[55] G. E. Bolton, E. Katok, and A. Ockenfels. Cooperation among strangers with limited information about reputation. *Journal of Public Economics*, 89(8):1457–1468, August 2005. 167

[56] P. Bourdieu. The forms of capital. *Handbook for Theory and Research for the Sociology of Education*, pages 241–258, 1986. 68, 160

[57] R. Boyd, H. Gintis, S. Bowles, and P. J. Richerson. The evolution of altruistic punishment. *Proceedings of the National Academy of Sciences of the USA*, March 2003. 157

[58] D. Braess. On a paradox of traffic planning. *Transportation Science*, 39(4):446–450, November 2005. 48

[59] S. Buchegger and J.-Y. LeBoudec. Performance analysis of the CONFIDANT protocol. In *MobiHoc '02: Proceedings of the 3rd ACM international symposium on Mobile ad hoc networking & computing*, pages 226–236, New York, NY, USA, 2002. ACM Press. 42

[60] S. Buchegger and J. Y. LeBoudec. The effect of rumor spreading in reputation systems for mobile ad-hoc networks. In *Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks (WiOpt)*, 2003. 41

[61] C. Buragohain, D. Agrawal, and S. Suri. A game theoretic framework for incentives in P2P systems. In *P2P '03: Proceedings of the 3rd International Conference on Peer-to-Peer Computing*, page 48, Washington, DC, USA, 2003. IEEE Computer Society. 9, 26

[62] R. Buyya, S. J. Chapin, and D. C. DiNucci. Architectural models for resource management in the grid. In *GRID '00: Proceedings of the First IEEE/ACM International Workshop on Grid Computing*, pages 18–35, London, UK, 2000. Springer-Verlag. 15

[63] X.-R. Cao, H.-X. Shen, R. Milito, and P. Wirth. Internet pricing with a game theoretical approach: concepts and examples. *IEEE/ACM Trans. Netw.*, 10(2):208–216, 2002. ISSN 1063-6692. 9, 49

[64] M. Castro, P. Druschel, Y. C. Hu, and A. Rowstron. Exploiting network proximity in distributed hash tables. In O. Babaoglu, K. Birman, and K. Marzullo, editors, *International Workshop on Future Directions in Distributed Computing (FuDiCo)*, pages 52–55, June 2002. 17, 20

[65] M. Castro, P. Druschel, A.-M. Kermarrec, A. Nandi, A. Rowstron, and A. Singh. Splitstream: High-bandwidth multicast in a cooperative environment. In *19th ACM Symposium on Operating Systems Principles (SOSP'03)*, Oct. 2003. 16, 19, 47

[66] M. Castro, P. Druschel, A.-M. Kermarrec, A. Nandi, A. Rowstron, and A. Singh. *Peer-to-Peer Systems II*, volume 2735/2003 of *Lecture Notes in Computer Science*, chapter SplitStream: High-Bandwidth Content Distribution in Cooperative Environments, pages 292–303. Springer Berlin / Heidelberg, 2003. 16, 19

[67] M. Castro, P. Druschel, A.-M. Kermarrec, and A. Rowstron. Scalable application-level anycast for highly dynamic groups. In *Networked Group Communication, Fifth International COST264 Workshop (NGC'2003)*, Sept. 2003. 17, 20

[68] D. Chaum. The dining cryptographers problem: Unconditional sender and recipient untraceability. *Journal of Cryptology*, 1(1):65–75, 1988. 169

[69] D. Chaum and T. P. Pedersen. Transferred cash grows in size. In *EUROCRYPT*, pages 390–407, 1992. 24, 25

[70] N. Chen, A. Ghosh, and N. Lambert. Social lending. In *Proceedings of the ACM Conference on Electronic Commerce (EC'09)*, 2009. 25, 150

[71] A. Cheng and E. Friedman. Sybilproof reputation mechanisms. In *Proceedings of P2PEcon '05*, pages 128–132, New York, USA, 2005. ACM. 10, 54, 55, 56, 70, 78, 150

[72] M. X. Cheng, Y. Li, and D.-Z. Du. *Combinatorial Optimization in Communication Networks*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006. ISBN 0387290257. 15

[73] V. L. R. Chinthalapati, N. Yadati, and R. Karumanchi. Learning dynamic prices in multiseller electronic retail markets with price sensitive customers, stochastic demands, and inventory replenishments. *IEEE Transactions on Systems, Man, and Cybernetics (Part C: Applications and Reviews)*, 36(1):92–106, 2006. 84

[74] S. C. Choi and R. Wette. Maximum likelihood estimation of the parameters of the gamma distribution and their bias. *Technometrics*, 11(4):683–690, November 1969. 118

[75] Y. Chu, S. Rao, S. Seshan, and H. Zhang. Enabling conferencing applications on the internet using an overlay multicast architecture. *SIGCOMM Comput. Commun. Rev.*, 31(4):55–67, 2001. 16

[76] Y. Chu, J. Chuang, and H. Zhang. A case for taxation in peer-to-peer streaming broadcast. In *PINS '04: Proceedings of the ACM SIGCOMM workshop on Practice and theory of incentives in networked systems*, pages 205–212, New York, NY, USA, 2004. ACM. ISBN 1-58113-942-9. 47

[77] J. Chuang. Designing incentive mechanisms for peer-to-peer systems. *Grid Economics and Business Models, 2004. GECON 2004. 1st IEEE International Workshop on*, pages 67–81, 23 April 2004. doi: 10.1109/GECON.2004.1317584. 4

[78] B. Chun, Y. Fu, and A. Vahdat. Bootstrapping a distributed computational economy. In *Workshop on Economics of Peer-to-Peer Systems*, 2003. 24, 25, 114

[79] B. N. Chun, P. Buonadonna, A. AuYoung, C. Ng, D. C. Parkes, J. Shneidman, A. C. Snoeren, and A. Vahdat. Mirage: A microeconomic resource allocation system for sensornet testbeds. In *Proceedings of 2nd IEEE Workshop on Embedded Networked Sensors (EmNetsII)*, 2005. 28

[80] D. J. Clark and C. Riis. Rank-order tournaments and selection. *Journal of Economics*, 73(2): 167–191, June 2001. 44

[81] E. H. Clarke. Multipart pricing of public goods. *Public Choice*, 11(1), September 1971. 48

[82] D. Cliff. Genetic optimization of adaptive trading agents for double-auction markets. *Computational Intelligence for Financial Engineering (CIFEr), 1998. Proceedings of the IEEE/IAFE/INFORMS 1998 Conference on*, pages 252–258, 29-31 Mar 1998. doi: 10.1109/CIFER.1998.690152. 72

[83] C. W. Cobb and P. H. Douglas. A theory of production. *The American Economic Review*, 18(1): 139–165, March 1928. 136, 150

[84] B. Cohen. Incentives build robustness in BitTorrent. In *Proceedings of the Workshop on Economics of Peer-to-Peer Systems (NetEcon '03)*, Berkeley, CA, USA, 2003. 6, 30, 69

[85] J. Coleman. *Foundations of Social Theory*. Harvard University Press, August, 1998. 159, 162

[86] J. S. Coleman. Social capital in the creation of human capital. *The American Journal of Sociology*, 94:S95–S120, 1988. 13, 14, 43, 68, 71, 122, 159, 160, 161

[87] A. R. Conn, N. I. M. Gould, and P. L. Toint. *Trust-region methods*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2000. 133

[88] W. Conner and K. Nahrstedt. Securing peer-to-peer media streaming systems from selfish and malicious behavior. In *MDS '07: Proceedings of the 4th on Middleware doctoral symposium*, pages 1–6, New York, NY, USA, 2007. ACM. 21

[89] V. D. M. Consultants. Final report: Broadband internet access cost (first half of 2008). Technical report, European Commision: Information Society and Media Directorate-General, 2008. 23

[90] R. Cox, F. Dabek, F. Kaashoek, J. Li, and R. Morris. Practical, distributed network coordinates. *SIGCOMM Comput. Commun. Rev.*, 34(1):113–118, 2004. 20, 105, 110

[91] P. Cramton, Y. Shoham, and R. Steinberg. *Combinatorial Auctions*. MIT Press, 2006. 27, 101

[92] E. Crosby. Fire prevention. *Annals of the American Academy of Political and Social Science*, 26 (Insurance):224–238, September 1905. 166

[93] J. Crowcroft. Net neutrality: the technical side of the debate: a white paper. *SIGCOMM Comput. Commun. Rev.*, 37(1):49–56, 2007. 49

[94] F. Dabek, R. Cox, F. Kaashoek, and R. Morris. Vivaldi: a decentralized network coordinate system. *SIGCOMM Comput. Commun. Rev.*, 34(4):15–26, 2004. 17, 20, 105, 110

[95] H. A. David and H. N. Nagaraja. *Order Statistics*. Wiley Series in Probability and Statistics. Wiley-Interscience, 2003. 90

[96] R. Dawkins. *The Selfish Gene*. Oxford University Press, September 1990. 157, 167

[97] C. Dellarocas. *Reputation Mechanisms*. Elsevier, 2006. 69, 166, 168

[98] H. Deshpande, M. Bawa, and H. Garcia-Molina. Streaming live media over a peer-to-peer network. Technical report, Stanford University, 2001. 16

[99] Z. Despotovic and K. Aberer. A Probabilistic Approach to Predict Peers' Performance in P2P Networks. In *CIA 2004*, Cooperative Information Agents, pages 62–76, 2004. best paper award Cooperative Information Agents VIII, 8th International Workshop, CIA 2004, Erfurt, Germany, September 27-29, 2004, Proceedings. Lecture Notes in Computer Science 3191 Springer 2004. 41

[100] R. Dingledine, M. Freedman, and D. Molnar. *Peer-to-Peer: Harnessing the Power of Disruptive Technologies*, chapter 16: Accountability. O'Reilly & Associates, Inc., Sebastopol, CA, USA, 2001. 15, 165, 168

[101] R. Dingledine, M. J. Freedman, and D. Molnar. The free haven project: distributed anonymous storage service. In *International workshop on Designing privacy enhancing technologies*, pages 67–95, New York, NY, USA, 2001. Springer-Verlag New York, Inc. 42

[102] R. Dingledine, N. Mathewson, and P. Syverson. Tor: The second-generation onion router. In *In Proceedings of the 13th USENIX Security Symposium*, pages 303–320, 2004. 33

[103] D. DiPalantino and R. Johari. Traffic engineering versus content distribution: A game theoretic perspective. In *The 28th IEEE International Conference on Computer Communications (INFOCOM 2009)*, Rio de Janeiro, Brazil, Brazil, 4 2009. 49

[104] M. Dischinger, A. Mislove, A. Haeberlen, and K. P. Gummadi. Detecting bittorrent blocking. In *IMC '08: Proceedings of the 8th ACM SIGCOMM conference on Internet measurement*, pages 3–8, New York, NY, USA, 2008. ACM. 49

[105] J. R. Douceur. The Sybil attack. In *IPTPS '01: Revised Papers from the First International Workshop on Peer-to-Peer Systems*, pages 251–260, London, UK, 2002. Springer-Verlag. ISBN 3-540-44179-4. 54, 112

[106] J. R. Douceur. The sybil attack. In *Proccedings of IPTPS 2002*, pages 251–260, Cambridge, MA, March 2002. 53, 54, 69, 78

[107] J. Elster. *Handbook on the Economics of Giving, Reciprocity and Altruism*, volume 1, chapter Chapter 3: Altruistic Behavior and Altruistic Motivations, pages 183–206. Elsevier, 2006. 157, 167

[108] P. T. Eugster, R. Guerraoui, A.-M. Kermarrec, and L. Massoulie. From Epidemics to Distributed Computing. *IEEE Computer*, 37(5):60–67, 2004. doi: NA. 19

[109] E. Fehr and S. Gachter. Altruistic punishment in humans. *Nature*, 415(6868):137–140, 2002. 157

[110] Z. Fei, S. Bhattacharjee, E. W. Zegura, and M. H. Ammar. A novel server selection technique for improving the response time of a replicated service. In *Proc. INFOCOM 1998 (2)*, pages 783–791, 1998. URL `citeseer.ist.psu.edu/article/fei98novel.html`. 20

[111] J. Feigenbaum and S. Shenker. Distributed algorithmic mechanism design: recent results and future directions. In *DIALM '02: Proceedings of the 6th international workshop on Discrete algorithms and methods for mobile computing and communications*, pages 1–13, New York, NY, USA, 2002. ACM Press. 2, 4, 15, 48, 113

[112] J. Feigenbaum, L. Fortnow, D. M. Pennock, and R. Sami. Computation in a distributed information market. In *EC '03: Proceedings of the 4th ACM conference on Electronic commerce*, pages 156–165. ACM, 2003. 125

[113] J. Feigenbaum, V. Ramachandran, and M. Schapira. Incentive-compatible interdomain routing. In *EC '06: Proceedings of the 7th ACM conference on Electronic commerce*, pages 130–139, New York, NY, USA, 2006. ACM. 49

[114] M. Feldman and J. Chuang. Overcoming free-riding behavior in peer-to-peer systems. *SIGecom Exch.*, 5(4):41–50, 2005. 15, 113

[115] M. Feldman, K. Lai, J. Chuang, and I. Stoica. Quantifying disincentives in peer-to-peer networks. *Workshop on Economics of Peer-to-Peer Systems*, June 2003. 16, 44

[116] M. Feldman, K. Lai, I. Stoica, and J. Chuang. Robust incentive techniques for peer-to-peer networks. In *EC '04: Proceedings of the 5th ACM conference on Electronic commerce*, pages 102–111, New York, NY, USA, 2004. ACM. 113

[117] M. Feldman, C. Papadimitriou, J. Chuang, and I. Stoica. Free-riding and whitewashing in peer-to-peer systems. In *PINS '04: Proceedings of the ACM SIGCOMM workshop on Practice and theory of incentives in networked systems*, pages 228–236, New York, NY, USA, 2004. ACM Press. 53, 112

[118] M. Feldman, J. Chuang, I. Stoica, and S. Shenker. Hidden-action in multi-hop routing. In *EC '05: Proceedings of the 6th ACM conference on Electronic commerce*, pages 117–126. ACM, 2005. 43, 149

[119] M. Feldman, K. Lai, and L. Zhang. A Price-Anticipating Resource Allocation Mechanism for Distributed Shared Clusters. In *Proceedings of the ACM Conference on Electronic Commerce*, June 2005. 27, 84

[120] M. Feldman, J. Chuang, I. Stoica, and S. Shenker. Hidden-action in network routing. *Selected Areas in Communications, IEEE Journal on*, 25(6):1161–1172, August 2007. 43, 149

[121] D. F. Ferguson, C. Nikolaou, J. Sairamesh, and Y. Yemini. Economic models for allocating resources in computer systems. *Market-based control: a paradigm for distributed resource allocation*, pages 156–183, 1996. 22, 114

[122] B. Ford, P. Srisuresh, and D. Kegel. Peer-to-peer communication across network address translators. In *ATEC05: Proceedings of the annual conference on USENIX Annual Technical Conference*, pages 13–13, Berkeley, USA, 2005. USENIX Association. 23

[123] C. Fragouli, J.-Y. LeBoudec, and J. Widmer. Network coding: an instant primer. *SIGCOMM Comput. Commun. Rev.*, 36(1):63–68, 2006. 47

[124] R. H. Frank. Melding sociology and economics: James coleman's foundations of social theory. *Journal of Economic Literature*, 30(1):147–70, March 1992. 157, 159, 162

[125] Y. Fu, J. Chase, B. Chun, S. Schwab, and A. Vahdat. SHARP: an architecture for secure resource peering. In *SOSP '03: Proceedings of the nineteenth ACM symposium on Operating systems principles*, pages 133–148, New York, NY, USA, 2003. ACM Press. 24, 25, 29

[126] F. Fukuyama. Social capital and the global economy. *Foreign Affairs*, 74(5):89–103, September 1995. 43, 159, 163

[127] F. Fukuyama. Social capital and civil society. *IMF Conference on Second Generation Reforms*, 1999. 68, 159, 161, 162, 163

[128] M. Gairing, B. Monien, and K. Tiemann. Selfish routing with incomplete information. In *SPAA '05: Proceedings of the seventeenth annual ACM symposium on Parallelism in algorithms and architectures*, pages 203–212, New York, NY, USA, 2005. ACM. 41, 49

[129] T. E. Gamal. A public key cryptosystem and a signature scheme based on discrete logarithms. In *Proceedings of CRYPTO 84 on Advances in cryptology*, pages 10–18, New York, NY, USA, 1985. Springer-Verlag New York, Inc. 165

[130] P. Garbacki, A. Iosup, D. Epema, and M. van Steen. 2Fast: Collaborative downloads in P2P networks (best paper award). In *6-th IEEE International Conference on Peer-to-Peer Computing*, pages 23–30. IEEE Computer Society, September 2006. 39

[131] P. Garbacki, D. H. J. Epema, and M. van Steen. An amortized Tit-For-Tat protocol for exchanging bandwidth instead of content in P2P networks. In *Proceedings of SASO 2007*, Boston, MA, July 2007. 25, 26

[132] F. D. Garcia and J.-H. Hoepman. Off-line Karma: A decentralized currency for peer-to-peer and grid applications. In J. Ioannidis, A. Keromytis, and M. Yung, editors, *3th Applied Cryptography and Network Security (ACNS 2005)*, volume 3531 of *Lecture Notes in Computer Science*, pages 364–377, New York, NY, U.S.A., June 7–10 2005. Springer Verlag. 8, 24, 69

[133] M. Garetto, D. R. Figueiredo, R. Gaeta, and M. Sereno. A modeling framework to understand the tussle between isps and peer-to-peer file-sharing users. *Perform. Eval.*, 64(9-12):819–837, 2007. 49

[134] S. Gessel. *The Natural Economic Order*. Peter Owen Limited, 1958. 158

[135] D. Ghosal, B. K. Poon, and K. Kong. P2p contracts: a framework for resource and service exchange. *Future Gener. Comput. Syst.*, 21(3):333–347, 2005. 43

[136] A. Ghosh, M. Mahdian, D. M. Reeves, D. M. Pennock, and R. Fugger. Mechanism design on trust networks. In *Third International Workshop on Internet and Network Economics*, volume 4858, pages 257–268, San Diego, CA, USA, December 2007. 4, 150

[137] H. Gintis. *Game Theory Evolving*. Princeton University Press, 2000. 58

[138] D. K. Gode and S. Sunder. Allocative efficiency of markets with zero-intelligence traders: Market as a partial substitute for individual rationality. *Journal of Political Economy*, 101(1):119–37, February 1993. 21

[139] S. Goldberg, S. Halevi, A. D. Jaggard, V. Ramachandran, and R. N. Wright. Rationality and traffic attraction: incentives for honest path announcements in bgp. In *SIGCOMM '08: Proceedings of the ACM SIGCOMM 2008 conference on Data communication*, pages 267–278, New York, NY, USA, 2008. ACM. 49

[140] S. Gollapudi, D. Sivakumar, and A. Zhang. Exploiting anarchy in networks: a game-theoretic approach to combining fairness and throughput. In *INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies, 13-17 March 2005, Miami, FL, USA*, pages 2147–2158, 2005. 48

[141] P. Golle, K. Leyton-Brown, and I. Mironov. Incentives for sharing in peer-to-peer networks. In *EC '01: Proceedings of the 3rd ACM conference on Electronic Commerce*, pages 264–267, New York, NY, USA, 2001. ACM Press. 9, 22, 26

[142] V. K. Goyal. Multiple description coding: Compression meets the network. *IEEE Signal Processing Magazine*, 18(5):74–93, September 2001. 45

[143] M. S. Granovetter. The strength of weak ties. *The American Journal of Sociology*, 78(6):1360–1380, 1973. 159, 164

[144] H. Gravelle and R. Rees. *Microeconomics*. Prentice Hall (Financial Times), 3rd edition, 2004. 3, 136, 141, 153

[145] B. Greiner and Vittoria. Indirect reciprocity in cyclical networks: An experimental study. *Journal of Economic Psychology*, 26(5):711–731, October 2005. 44

[146] T. G. Griffin and J. L. Sobrinho. Metarouting. In *Proceedings of SIGCOMM '05*, pages 1–12, New York, NY, USA, 2005. ACM. ISBN 1-59593-009-4. doi: http://doi.acm.org/10.1145/1080091.1080094. 75, 150

[147] D. Grolimund, L. Meisser, S. Schmid, and R. Wattenhofer. Havelaar: A Robust and Efficient Reputation System for Active Peer-to-Peer Systems. In *1st Workshop on the Economics of Networked Systems (NetEcon), University of Michigan, Ann Arbor, Michigan, USA*, June 2006. 41, 53

[148] J. Gross and J. Yellen. *Graph Theory and its Applications*. CRC Press, 1999. 51

[149] C. Grothoff. An Excess-Based Economic Model for Resource Allocation in Peer-to-Peer Networks. *Wirtschaftsinformatik*, 3-2003, June 2003. 28, 29, 31, 45, 88

[150] T. Groves. Incentives in teams. *Econometrica*, 41(4):617–31, July 1973. 48

[151] S. Guha and P. Francis. Characterization and measurement of TCP traversal through NATs and firewalls. In *IMC05: Proceedings of the Internet Measurement Conference 2005 on Internet Measurement Conference*, pages 18–18, Berkeley, USA, 2005. USENIX Association. 23

[152] L. Guo, S. Chen, Z. Xiao, E. Tan, X. Ding, and X. Zhang. Measurements, analysis, and modeling of BitTorrent-like systems. In *IMC '05, Internet Measurement Conference*, pages 35–48, 2005. 34, 35

[153] R. Gupta and A. K. Somani. Reputation management framework and its use as currency in large-scale peer-to-peer networks. In *Proceedings of P2P'04*, pages 124–132, Washington, DC, USA, 2004. IEEE Computer Society. 42, 69

[154] R. Gupta, V. Sekhri, and A. K. Somani. CompuP2P: An architecture for internet computing using peer-to-peer networks. *IEEE Transactions on Parallel and Distributed Systems*, 17(11):1306–1320, 2006. ISSN 1045-9219. 27

[155] A. Habib and J. C.-I. Chuang. Service differentiated peer selection: an incentive mechanism for peer-to-peer media streaming. *IEEE Transactions on Multimedia*, 8(3):610–621, 2006. 44, 45

[156] D. Hales and S. Arteconi. SLACER: a self-organizing protocol for coordination in peer-to-peer networks. *Intelligent Systems, IEEE*, 21(2):29–35, March-April 2006. 43

[157] M. Handley, S. Floyd, J. Padhye, and J. Widmer. RFC 3448 - TCP friendly rate control (TFRC): Protocol specification, 2003. 117

[158] G. Hardin. The tragedy of the commons. *Science*, 162:1243–1248, 1968. 14

[159] N. Hardy and E. D. Tribble. The digital silk road. *Agoric Systems: Market Based Computation*, September 1995. URL `http://www.agorics.com/Library/dsr.html`. 22, 29

[160] J. D. Hartline and T. Roughgarden. Optimal mechanism design and money burning. In *STOC '08: Proceedings of the 40th annual ACM symposium on Theory of Computing*, pages 75–84, New York, NY, USA, 2008. ACM. 4, 64

[161] D. Hausheer. *PeerMart: Secure Decentralized Pricing and Accounting for Peer-to-Peer Systems*. Ph.d., ETH Zurich, Aachen, Germany, March 2006. 72, 84

[162] F. A. Hayek. The use of knowledge in society. *The American Economic Review*, 35(4):519–530, 1945. 156

[163] M. Hefeeda, A. Habib, B. Botev, D. Xu, and B. Bhargava. Promise: peer-to-peer media streaming using collectcast. In *MULTIMEDIA '03: Proceedings of the eleventh ACM international conference on Multimedia*, pages 45–54, New York, NY, USA, 2003. ACM Press. ISBN 1-58113-722-2. 16, 19, 44

[164] J. Horrigan. Home broadband adoption 2006. Technical report, Pew Internet and American Life Project, May 2006. 23

[165] J. Horrigan. Home broadband adoption 2008. Technical report, Pew Internet and American Life Project, April 2008. 23

[166] Y. hua Chu, S. G. Rao, S. Seshan, and H. Zhang. A Case for End-System Multicast. *IEEE Journal on Selected Areas in Communications special issue on Network Support for Multicast Communications*, 20(8), Oct. 2002. 16, 19

[167] D. Hughes, G. Coulson, and J. Walkerdine. Free riding on Gnutella revisited: The bell tolls? *IEEE Distributed Systems Online*, 6(6):1, 2005. 3, 14, 16

[168] L. Hurwicz and S. Reiter. Transversals, systems of distinct representatives, mechanism design, and matching. *Review of Economic Design*, 6(2):289–304, September 2001. 150

[169] L. Hurwicz and S. Reiter. *Designing Economic Mechanisms*. Cambridge University Press, May 2006. 64, 96, 150, 154

[170] A. Iosup, P. Garbacki, J. Pouwelse, and D. Epema. Correlating topology and path characteristics of overlay networks and the internet. In *CCGRID '06: Proceedings of the Sixth IEEE International Symposium on Cluster Computing and the Grid (CCGRID'06)*, page 10, Washington, DC, USA, 2006. IEEE Computer Society. 37, 38

[171] D. Irwin, J. Chase, L. Grit, and A. Yumerefendi. Self-recharging virtual currency. In *P2PECON '05: Proceeding of the 2005 ACM SIGCOMM workshop on Economics of peer-to-peer systems*, pages 93–98, New York, NY, USA, 2005. ACM Press. 29

[172] M. Izal, G. Urvoy-Keller, E. W. Biersack, P. A. Felber, A. A. Hamra, and L. Garces-Erice. Dissecting bittorrent: Five months in a torrent's lifetime. In *Proceedings of Passive and Active Measurements (PAM)*, volume 3015 of *Lecture Notes in Computer Science*, pages 1–11. April, Springer, 2004. 35, 37

[173] R. Izumi. The WAT system: An exchange system based on mutual appreciation, 2001. URL `http://home.debitel.net/user/RMittelstaedt/Money/watto-e.htm`. 24, 25, 28

[174] M. O. Jackson. A crash course in implementation theory. *Social Choice and Welfare*, 18(4): 655–708, 2001. 4, 64

[175] J. Jannotti, D. K. Gifford, K. L. Johnson, M. F. Kaashoek, and J. W. O'Toole, Jr. Overcast: reliable multicasting with on overlay network. In *OSDI'00: Proceedings of the 4th conference on Symposium on Operating System Design & Implementation*, pages 14–14, Berkeley, CA, USA, 2000. USENIX Association. 16

[176] M. Jelasity, R. Guerraoui, A.-M. Kermarrec, and M. van Steen. The peer sampling service: experimental evaluation of unstructured gossip-based implementations. In *Middleware '04: Proceedings of the 5th ACM/IFIP/USENIX international conference on Middleware*, pages 79–98, 2004. 125

[177] G. P. Jesi, A. Montresor, and Ö. Babaoglu. Proximity-aware superpeer overlay topologies. In *Second IEEE International Workshop in Self-Managed Networks, Systems, and Services, SelfMan 2006*, pages 43–57, Dublin, Ireland, June 2006. 148

[178] L. Jian and J. MacKie-Mason. Why Share in Peer-to-Peer Networks? In *1st Workshop on the Economics of Networked Systems (NetEcon), University of Michigan, Ann Arbor, Michigan, USA*, May 2006. 44

[179] J. Jiang, H. Bai, and W. Wang. *Trust and Cooperation in Peer-to-Peer Systems*, volume 3032/2004 of *Lecture Notes in Computer Science*, pages 371 – 378. Springer Berlin / Heidelberg, 2004. 166

[180] W. Jiang, R. Zhang-Shen, J. Rexford, and M. Chiang. Cooperative content distribution and traffic engineering. In *NetEcon '08: Proceedings of the 3rd international workshop on Economics of networked systems*, pages 7–12. ACM, 2008. 49

[181] S. Jun and M. Ahamad. Incentives inBitTorrentinduce free riding. In *P2PECON '05: Proceeding of the 2005 ACM SIGCOMM workshop on Economics of peer-to-peer systems*, pages 116–121, New York, NY, USA, 2005. ACM Press. ISBN 1-59593-026-4. doi: http://doi.acm.org/10.1145/1080192.1080199. 15

[182] D. Jurca, J. Chakareski, J.-P. Wagner, and P. Frossard. Enabling adaptive video streaming in P2P systems [peer-to-peer multimedia streaming]. *Communications Magazine, IEEE*, 45(6):108–114, June 2007. 16

[183] M. A. Kaafar, L. Mathy, C. Barakat, K. Salamatian, T. Turletti, and W. Dabbous. Securing internet coordinate embedding systems. *SIGCOMM Comput. Commun. Rev.*, 37(4):61–72, 2007. 53

[184] N. Kamiyama, T. Mori, R. Kawahara, S. Harada, and H. Hasegawa. ISP-Operated CDN. In *12th IEEE Global Internet Symposium (INFOCOM workshops)*, Rio de Janeiro, Brazil, April 2009. 49

[185] S. D. Kamvar, M. T. Schlosser, and H. Garcia-Molina. The EigenTrust algorithm for reputation management in P2P networks. In *Proceedings of WWW '03*, pages 640–651, New York, USA, 2003. ACM Press. 40, 55, 69

[186] S. D. Kamvar, M. T. Schlosser, and H. Garcia-Molina. Incentives for combatting freeriding on P2P networks. In *Parallel Processing, 9th International Euro-Par Conference*, volume 2790 of *Lecture Notes in Computer Science*, pages 1273–1279, Klagenfurt, Austria, August 26-29 2003. 15, 40

[187] M. Kandori. Social norms and community enforcement. *Review of Economic Studies*, 59(1): 63–80, January 1992. 41, 42, 122, 168

[188] D. Karger, A. Sherman, A. Berkheimer, B. Bogstad, R. Dhanidina, K. Iwamoto, B. Kim, L. Matkins, and Y. Yerushalmi. Web caching with consistent hashing. In *WWW '99: Proceedings of the eighth international conference on World Wide Web*, pages 1203–1213, New York, NY, USA, 1999. Elsevier North-Holland, Inc. 17

[189] D. Katabi and J. Wroclawski. A framework for scalable global IP-anycast (GIA). In *Proc. SIGCOMM 2000*, pages 3–15, 2000. URL `citeseer.ist.psu.edu/article/katabi00framework.html`. 20

[190] G. Kendall, X. Yao, and S. Y. Chong. *The Iterated Prisoner's Dilemma: 20 years on*. WSPC, 2007. 77

[191] B. Khorshadi, X. Liu, and D. Ghosal. Determining the peer resource contributions in a P2P contract. *Hot Topics in Peer-to-Peer Systems, 2005. HOT-P2P 2005. Second International Workshop on*, pages 2–9, 21 July 2005. 43

[192] M. Klafft. Peer to peer lending: Auctioning microcredits over the internet. In *Proceedings of the 2nd International Conference on Information Systems, Technology and Management, ICISTM 2008*, Dubai, March 2008. 25

[193] Y. Korilis, A. Lazar, and A. Orda. Achieving network optima using stackelberg routing strategies. *Networking, IEEE/ACM Transactions on*, 5(1):161–173, Feb 1997. ISSN 1063-6692. doi: 10.1109/90.554730. 49

[194] D. Kostić, A. Rodriguez, J. Albrecht, and A. Vahdat. Bullet: high bandwidth data dissemination using an overlay mesh. In *SOSP '03: Proceedings of the nineteenth ACM symposium on Operating systems principles*, pages 282–297, New York, NY, USA, 2003. ACM. ISBN 1-58113-757-5. doi: http://doi.acm.org/10.1145/945445.945473. 16, 117

[195] G. Kwon and K. D. Ryu. An efficient peer-to-peer file sharing exploiting hierarchy and asymmetry. In *SAINT '03: Proceedings of the 2003 Symposium on Applications and the Internet*, page 226, Washington, DC, USA, 2003. IEEE Computer Society. 15

[196] J.-J. Laffont and D. Martimort. *The Theory of Incentives: The Principal-Agent Model*. Princeton University Press, 2001. 151

[197] K. Lai and M. Baker. Measuring bandwidth. In *INFOCOM*, pages 235–245, 1999. 24

[198] K. Lai, M. Feldman, I. Stoica, and J. Chuang. Incentives for cooperation in peer-to-peer networks. In *Proceedings of P2PEcon*, 2003. 29, 40, 68, 69, 166, 168

[199] D. Lal. Policy forum: Culture, democracy, and development. *Cato Policy Report*, 22(1), February 2000. 155, 157

[200] J. Ledlie, P. Gardner, and M. I. Seltzer. Network coordinates in the wild. In *NSDI*. USENIX, 2007. 17, 21

[201] D. Levin, K. LaCurts, N. Spring, and B. Bhattacharjee. BitTorrent is an auction: analyzing and improving BitTorrent's incentives. *SIGCOMM Comput. Commun. Rev.*, 38(4):243–254, 2008. 6, 27, 97

[202] H. Levin, M. Schapira, and A. Zohar. Interdomain routing and games. In *STOC '08: Proceedings of the 40th annual ACM symposium on Theory of computing*, pages 57–66, New York, NY, USA, 2008. ACM. 49

[203] B. N. Levine, C. Shields, and N. B. Margolin. A Survey of Solutions to the Sybil Attack. Tech report 2006-052, University of Massachusetts Amherst, Amherst, MA, October 2006. 3, 7, 54

[204] Z. Li and P. Mohapatra. QRON: QoS-aware routing in overlay networks. *IEEE Journal on Selected Areas in Communications*, 22(1):29–40, January 2004. 112

[205] Y.-N. Lien. Performance issues of P2P file sharing over asymmetric and wireless networks file sharing over asymmetric and wireless networks. In *ICDCSW '05: Proceedings of the First International Workshop on Mobility in Peer-to-Peer Systems (MPPS) (ICDCSW'05)*, pages 850–855, Washington, DC, USA, 2005. IEEE Computer Society. 37

[206] Lietaer. Complementary currencies in japan today: History, originality, relevance. *International Journal of Community Currency Research*, 7, 2004. 25, 68

[207] N. Liogkas, R. Nelson, E. Kohler, , and L. Zhang. Exploiting BitTorrent for fun (but not profit). In *Proceedings of the 5th International Workshop on Peer-to-Peer Systems (IPTPS'06)*, Santa Barbara, CA, USA, February 2006. 32, 33

[208] Y. Liu, Y. Guo, and C. Liang. A survey on peer-to-peer video streaming systems. *Peer-to-Peer Networking and Applications*, 1(1):18–28, March 2008. 16, 17

[209] T. Locher, P. Moor, S. Schmid, and R. Wattenhofer. Free riding in BitTorrent is cheap. In *5th Workshop on Hot Topics in Networks (HotNets), Irvine, California, USA*, November 2006. 15, 32, 33, 36

[210] K. Lua, J. Crowcroft, M. Pias, R. Sharma, and S. Lim. A survey and comparison of peer-to-peer overlay network schemes. *Communications Surveys & Tutorials, IEEE*, pages 72–93, March 2004. 17, 148

[211] R. T. Ma, D.-m. Chiu, J. C. S. Lui, V. Misra, and D. Rubenstein. Interconnecting eyeballs to content: a shapley value perspective on isp peering and settlement. In *NetEcon '08: Proceedings of the 3rd international workshop on Economics of networked systems*, pages 61–66, New York, NY, USA, 2008. ACM. 9, 49

[212] R. T. B. Ma, D. m. Chiu, J. C. S. Lui, V. Misra, and D. Rubenstein. Internet economics: the use of shapley value for isp settlement. In *CoNEXT '07: Proceedings of the 2007 ACM CoNEXT conference*, pages 1–12, New York, NY, USA, 2007. ACM. 49

[213] N. Magharei and R. Rejaie. Adaptive receiver-driven streaming from multiple senders. *Multimedia Systems*, 11(18):550–567, June 2006. 16

[214] N. Magharei and R. Rejaie. PRIME: Peer-to-peer receiver-driven mesh-based streaming. *INFOCOM 2007. 26th IEEE International Conference on Computer Communications. IEEE*, pages 1415–1423, May 2007. ISSN 0743-166X. 16

[215] N. Magharei, R. Rejaie, and Y. Guo. Mesh or multiple-tree: A comparative study of live P2P streaming approaches. *INFOCOM 2007. 26th IEEE International Conference on Computer Communications. IEEE*, pages 1424–1432, May 2007. 16

[216] G. J. Mailath and L. Samuelson. *Repeated Games and Reputations*. Cambridge University Press, Sept. 2006. 40

[217] A. Majumder, N. Shrivastava, R. Rastogi, and A. Srinivasan. Scalable content-based routing in Pub/Sub systems. In *The 28th IEEE International Conference on Computer Communications (INFOCOM 2009)*, Rio de Janeiro, Brazil, Brazil, 4 2009. 20, 125

[218] D. W. Marquardt. An algorithm for least-squares estimation of nonlinear parameters. *Journal of the Society for Industrial and Applied Mathematics*, 11(2):431–441, 1963. 144

[219] S. Marti and H. Garcia-Molina. Identity crisis: Anonymity vs. reputation in P2P systems. In *P2P '03: Proceedings of the 3rd International Conference on Peer-to-Peer Computing*, page 134, Washington, DC, USA, 2003. IEEE Computer Society. ISBN 0-7695-2023-5. 29, 54

[220] E. Maskin and T. Sjostrom. *Implementation theory*, chapter 5, pages 237–288. Elsevier, 2002. 4, 64

[221] L. Massoulié and M. Vojnović. Coupon replication systems. In *SIGMETRICS '05: Proceedings of the 2005 ACM SIGMETRICS international conference on Measurement and modeling of computer systems*, pages 2–13, New York, NY, USA, 2005. ACM Press. 36

[222] A. S. Matveev and A. V. Savkin. *Estimation and Control over Communication Networks (Control Engineering)*. Birkhauser, 2007. 151

[223] D. Mazières and M. F. Kaashoek. Escaping the evils of centralized control with self-certifying pathnames. In *EW 8: Proceedings of the 8th ACM SIGOPS European workshop on Support for composing distributed applications*, pages 118–125, New York, NY, USA, 1998. ACM. 70

[224] R. P. McAfee and J. McMillan. Game theory and competition. Working paper, University of Texas, 1998. 21

[225] J. McCoy and B. Wilcox. MojoNation, March 2007. URL `http://web.archive.org/web/{*}/www.mojonation.net`. 22

[226] A. J. Menezes. *Elliptic Curve Public Key Cryptosystems*. Kluwer Academic Publishers, Norwell, MA, USA, 1994. ISBN 0792393686. Foreword By-Neal Koblitz. 165

[227] F. M. Menezes and P. K. Monteiro. *An Introduction to Auction Theory*. Oxford University Press, USA, 1st edition, January 2005. 28, 64

[228] J. A. Mirrlees. The theory of moral hazard and unobservable behaviour: Part i. *Review of Economic Studies*, 66(1):3–21, January 1999. 115

[229] T. Moore. Countering hidden action attacks on networked systems. *Fourth Workshop on the Economics and Information Security*, June 2005. 43, 149

[230] R. Morris. TCP behavior with many flows. In *ICNP '97: Proceedings of the 1997 International Conference on Network Protocols (ICNP '97)*, page 205, Washington, DC, USA, 1997. IEEE Computer Society. 32, 33

[231] R. Morris, D. Karger, F. Kaashoek, and H. Balakrishnan. Chord: A Scalable Peer-to-Peer Lookup Service for Internet Applications. In *ACM SIGCOMM 2001*, San Diego, CA, September 2001. 17, 27, 41

[232] T. Moscibroda and S. Schmid. Mechanism design without payments for throughput maximization. In *The 28th IEEE International Conference on Computer Communications (INFOCOM 2009)*, Rio de Janeiro, Brazil, Brazil, 4 2009. 4, 64

[233] L. Mui, M. Mohtashemi, and A. Halberstadt. A computational model of trust and reputation. *Proceedings of the 35th Annual Hawaii International Conference on System Sciences (HICSS)*, pages 2431– 2439, January 2002. 166

[234] A. Mukherjee. On the dynamics and significance of low frequency components of internet load. *Internetworking: Research and Experience*, 5:163–205, December 1994. 118

[235] E. Mykoniati, L. Latif, R. Landa, B. Yang, R. Clegg, D. Griffin, and M. Rio. Distributed overlay anycast tables using space filling curves. In *12th IEEE Global Internet Symposium 2009*, Rio de Janeiro, Brazil, April 2009. 125

[236] J. Nash. Non-cooperative games. *Annals of Mathematics*, 54:286–295, 1951. 48, 57

[237] J. F. Nash. Equilibrium points in n-person games. *Proceedings of the National Academy of Sciences of the United States of America*, 36(1):48–49, 1950. 48

[238] J. V. Neumann and O. Morgenstern. *Theory of Games and Economic Behavior*. Princeton University Press, 1944. 57

[239] T. S. E. Ng and H. Zhang. A network positioning system for the internet. In *ATEC '04: Proceedings of the annual conference on USENIX Annual Technical Conference*, pages 11–11, Berkeley, CA, USA, 2004. USENIX Association. 17

[240] N. Nisan. Algorithms for selfish agents. In *Proc. STACS (Symposium on Theoretical Aspects of Computer Science)*, volume 1563 of *Lecture Notes in Computer Science*, pages 1–15, March 1999. 49

[241] N. Nisan and A. Ronen. Algorithmic mechanism design. *Games and Economic Behavior*, 35: 166–196, 2001. 2, 4, 15, 26, 48, 64

[242] N. Nisan, T. Roughgarden, E. Tardos, and V. V. Vazirani. *Algorithmic Game Theory*. Cambridge University Press, New York, NY, USA, 2007. 4, 41, 48, 54, 59, 64, 149

[243] M. A. Nowak and K. Sigmund. Evolution of indirect reciprocity by image scoring. *Nature*, 393 (6685):573–577, June 1998. 42, 43, 44

[244] M. A. Nowak and K. Sigmund. Evolution of indirect reciprocity. *Nature*, 437:1291–1298, October 2005. 39

[245] M. J. Osborne. *An Introduction to Game Theory*. Oxford University Press, USA, 2003. 58

[246] J. Padhye, V. Firoiu, D. Towsley, and J. Kurose. Modeling TCP throughput: a simple model and its empirical validation. *SIGCOMM Comput. Commun. Rev.*, 28(4):303–314, 1998. 119, 120

[247] V. N. Padmanabhan, H. J. Wang, and P. A. Chou. Resilient peer-to-peer streaming. In *ICNP '03: Proceedings of the 11th IEEE International Conference on Network Protocols*, page 16, Washington, DC, USA, 2003. IEEE Computer Society. 16, 19, 45, 46, 47

[248] V. Pai and A. E. Mohr. Improving robustness of peer-to-peer streaming with incentives. In *1st Workshop on the Economics of Networked Systems (NetEcon), University of Michigan, Ann Arbor, Michigan, USA*, June 2006. 6, 29, 45, 47

[249] C. Papadimitriou. Algorithms, games, and the internet. In *STOC '01: Proceedings of the thirty-third annual ACM symposium on Theory of computing*, pages 749–753, New York, NY, USA, 2001. ACM Press. 22, 48

[250] M. Pastore. U.S. residential internet market grows in second quarter, August 2001. URL `http://www.isp-planet.com/research/2001/us_q2.html`. 23

[251] R. Penrose. A generalized inverse for matrices. In *Proceedings of the Cambridge Philosophical Society*, volume 51, pages 406–413, 1955. 138

[252] J. Peterson, E. Marocco, and V. Gurbani. Application-Layer Traffic Optimization (ALTO) working group, 2009. URL `http://www.ietf.org/dyn/wg/charter/alto-charter.html`. 9, 49, 126

[253] F. Pianese, J. Keller, and E. W. Biersack. PULSE, a flexible P2P live streaming system. *INFOCOM 2006. 25th IEEE International Conference on Computer Communications. Proceedings*, pages 1–6, April 2006. ISSN 0743-166X. doi: 10.1109/INFOCOM.2006.42. 16, 19, 117

[254] M. Piatek, T. Isdal, T. Anderson, A. Krishnamurthy, and A. Venkataramani. Do incentives build robustness in BitTorrent? In *Proceedings of NSDI'07*, Cambridge, MA, April 2007. 30, 31, 36, 68

[255] J. Pouwelse, P. Garbacki, J. Wang, A. Bakker, J. Yang, A. Iosup, D. H. J. Epema, M. Reinders, M. van Steen, and H. Sips. Tribler: A social-based peer-to-peer system. *Concurrency and Computation: Practice and Experience*, 2007. 7, 39, 82

[256] J. A. Pouwelse, P. Garbacki, D. H. J. Epema, and H. J. Sips. The Bittorrent P2P file-sharing system: Measurements and analysis. In *4th Int'l Workshop on Peer-to-Peer Systems (IPTPS)*, volume 3640. LNCS, Feb 2005. 23, 35, 37, 45

[257] R. D. Putnam. *Bowling Alone : The Collapse and Revival of American Community*. Simon & Schuster, August 2001. 160, 162, 163, 164

[258] D. Qiu and R. Srikant. Modeling and performance analysis of BitTorrent-like peer-to-peer networks. In *SIGCOMM '04: Proceedings of the 2004 conference on Applications, technologies, architectures, and protocols for computer communications*, pages 367–378, New York, NY, USA, 2004. ACM Press. 36, 37

[259] L. Qiu, Y. R. Yang, Y. Zhang, and S. Shenker. On selfish routing in internet-like environments. *IEEE/ACM Trans. Netw.*, 14(4):725–738, 2006. 41, 49

[260] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Schenker. A scalable content-addressable network. In *SIGCOMM '01: Proceedings of the 2001 conference on Applications, technologies, architectures, and protocols for computer communications*, pages 161–172, New York, NY, USA, 2001. ACM. ISBN 1-58113-411-8. doi: http://doi.acm.org/10.1145/383059.383072. 17

[261] S. Reiter and L. Hurwicz. On transversals and systems of distinct representatives. Technical Report 1176R, Center for Mathematical Studies in Economics and Management Science, Northwestern University,, 580 Jacobs Center, 2001 Sheridan Road, Evanston, IL 60208-2014, January 1997. 150

[262] P. Resnick and R. Sami. Sybilproof transitive trust protocols. In *EC '09: Proceedings of the tenth ACM conference on Electronic commerce*, pages 345–354, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-458-4. doi: http://doi.acm.org/10.1145/1566374.1566423. 3, 10, 150

[263] P. Resnick, K. Kuwabara, R. Zeckhauser, and E. Friedman. Reputation systems. *Commun. ACM*, 43(12):45–48, 2000. 40

[264] P. Resnick, R. Zeckhauser, J. Swanson, and K. Lockwood. The value of reputation on eBay: A controlled experiment. *Experimental Economics*, 9(2):79–101, June 2006. 168

[265] S. C. Rhea and J. Kubiatowicz. Probabilistic location and routing. *INFOCOM 2002. Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings.*, 3: 1248–1257, 2002. 20, 40

[266] R. L. Rivest and C. E. Leiserson. *Introduction to Algorithms*. McGraw-Hill, Inc., New York, NY, USA, 1990. ISBN 0070131430. 75

[267] R. L. Rivest, A. Shamir, and L. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, 26(1):96–99, 1983. 165

[268] S. Ross. *A First Course in Probability (8th Edition)*. Prentice Hall, 8 edition, November 2008. 119

[269] T. Roughgarden. *Selfish Routing and the Price of Anarchy*. The MIT Press, 2005. 41

[270] A. Rowstron and P. Druschel. Pastry: Scalable, decentralized object location and routing for large-scale peer-to-peer systems. In *IFIP/ACM International Conference on Distributed Systems Platforms (Middleware)*, pages 329–350, Nov. 2001. 17, 20, 23

[271] A. Rowstron, A.-M. Kermarrec, M. Castro, and P. Druschel. Scribe: The design of a large-scale event notification infrastructure. In J. Crowcroft and M. Hofmann, editors, *Networked Group Communication, Third International COST264 Workshop (NGC'2001)*, volume 2233 of *Lecture Notes in Computer Science*, pages 30–43, Nov. 2001. 17, 20

[272] D. Rubenstein, J. Kurose, and D. Towsley. Detecting shared congestion of flows via end-to-end measurement. *IEEE/ACM Trans. Netw.*, 10(3):381–395, 2002. 145

[273] K. Saito. *i-WAT: The Internet WAT System – An Architecture for Maintaining Trust and Facilitating Peer-to-Peer Barter Relationships*. PhD thesis, Graduate School of Media and Governance, Keio University, February 2006. 24, 25, 28, 29, 68, 74

[274] K. Saito, E. Morino, and J. Murai. Fair trading of information: A proposal for the economics of peer-to-peer systems. In *ARES '06: Proceedings of the First International Conference on Availability, Reliability and Security (ARES'06)*, pages 764–771, Washington, DC, USA, 2006. IEEE Computer Society. 29

[275] K. Saito, E. Morino, Y. Suko, T. Suzuki, and J. Murai. Local production, local consumption peer-to-peer architecture for a dependable and sustainable social infrastructure. *International Symposium on Applications and the Internet Workshops (SAINT)*, January 2007. 29

[276] S. Sanghavi and B. Hajek. A new mechanism for the free-rider problem. In *P2PECON '05: Proceeding of the 2005 ACM SIGCOMM workshop on Economics of peer-to-peer systems*, pages 122–127, New York, NY, USA, 2005. ACM Press. 15, 49, 113

[277] S. Saroiu, P. K. Gummadi, and S. Gribble. A measurement study of peer-to-peer file sharing systems. In *SPIE Multimedia Computing and Networking (MMCN2002)*, 2002. 15, 16

[278] B. Schneier. *Applied Cryptography: Protocols, Algorithms, and Source Code in C*. John Wiley & Sons, Inc., New York, USA, 1995. 165

[279] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson. RFC 3550 - RTP: A transport protocol for real-time applications, 2003. 117

[280] Y. Shavitt and T. Tankel. The curvature of the internet and its usage for overlay construction and distance estimation. In *NFOCOM 2004. Twenty-third AnnualJoint Conference of the IEEE Computer and Communications Societies*, volume 1, pages – 384. Proc. of IEEE Infocom, April 2004., 2004. 17, 20

[281] R. Sherwood, S. Lee, and B. Bhattacharjee. Cooperative peer groups in NICE. *Comput. Networks*, 50(4):523–544, 2006. ISSN 1389-1286. doi: http://dx.doi.org/10.1016/j.comnet.2005.07.012. 16, 19, 40, 41

[282] S. Sheu, K. Hua, and W. Tavanapong. Chaining: a generalized batching technique for video-on-demand systems. *Proc. IEEE Int. Conf. Multimedia Comp. Sys.*, pages 110–117, 1997. 16, 19

[283] J. Shneidman and D. C. Parkes. Rationality and self-interest in peer to peer networks. In *2nd Int. Workshop on Peer-to-Peer Systems (IPTPS'03)*, 2003. 49, 153

[284] J. Shneidman, C. Ng, D. C. Parkes, A. AuYoung, A. C. Snoeren, A. Vahdat, , and B. Chun. Why markets could (but don't currently) solve resource allocation problems in systems. In *Proceedings of Tenth Workshop on Hot Topics in Operating Systems*, Santa Fe, NM, USA, June 2005. USENIX - (TCOS). 21

[285] H. A. Simon. A behavioral model of rational choice. *The Quarterly Journal of Economics*, 69(1): 99–108, 1955. 167

[286] M. Sirivianos, J. H. Park, R. Chen, and X. Yang. Free-riding in BitTorrent networks with the large view exploit. In *Proc. IPTPS*, 2007. 32

[287] L. D. Solomon. *Rethinking our Centralized Monetary System: The Case for a System of Local Currencies*. Praeger Publishers, 1996. 25, 68

[288] I. Stoica, D. Adkins, S. Zhuang, S. Shenker, and S. Surana. Internet indirection infrastructure. *IEEE/ACM Transactions on Networking (TON)*, 12(2):205–218, April 2004. 20

[289] M. Stumm. Strategies for decentralized resource management. In *SIGCOMM '87: Proceedings of the ACM workshop on Frontiers in computer communications technology*, pages 245–253, New York, NY, USA, 1988. ACM Press. 15

[290] D. Stutzbach and R. Rejaie. Understanding churn in peer-to-peer networks. In *Proceedings of IMC '06*, pages 189–202, New York, USA, 2006. ACM. 23, 35, 166

[291] A.-J. Su, D. R. Choffnes, A. Kuzmanovic, and F. E. Bustamante. Drafting behind akamai. *ACM SIGCOMM Computer Communication Review*, 36:435 – 446, August 2006. 20

[292] L. Subramanian, I. Stoica, H. Balakrishnan, and R. Katz. OverQoS: offering internet QoS using overlays. In *ACM SIGCOMM CCR*, volume 33, pages 11–16, January 2003. 112

[293] J. Surowiecki. *The wisdom of crowds*. Abacus, 2004. 21

[294] I. E. Sutherland. A futures market in computer time. *Commun. ACM*, 11(6):449–451, 1968. 21, 29

[295] K. Tamilman, V. Pai, and A. Mohr. SWIFT: A system with incentives for trading. In *Proceedings of Second Workshop of Economics in Peer-to-Peer Systems*, 2004. 25, 114

[296] G. Tan and S. A. Jarvis. A payment-based incentive and service differentiation mechanism for peer-to-peer streaming broadcast. *14th IEEE International Workshop on Quality of Service (IWQoS)*, pages 41–50, June 2006. 44

[297] W. W. Terpstra, J. Kangasharju, C. Leng, and A. P. Buchmann. Bubblestorm: resilient, probabilistic, and exhaustive peer-to-peer search. *SIGCOMM Comput. Commun. Rev.*, 37(4):49–60, 2007. 148

[298] R. Thommes and M. Coates. Bittorrent fairness: analysis and improvements. In *Proc. Workshop Internet, Telecom. and Signal Proc.*, Australia, Dec. 2005. 30

[299] R. L. Trivers. The evolution of reciprocal altruism. *Quarterly Review of Biology*, 46:35, 1971. 71, 157, 167

[300] D. Tsang, K. Ross, P. Rodriguez, J. Li, and G. Karlsson. Advances in peer-to-peer streaming systems [guest editorial]. *Selected Areas in Communications, IEEE Journal on*, 25(9):1609–1611, December 2007. ISSN 0733-8716. doi: 10.1109/JSAC.2007.071201. 16

[301] Y. Tsang, M. Coates, and R. Nowak. Network delay tomography. *Signal Processing, IEEE Transactions on*, 51(8):2125–2136, Aug. 2003. 145

[302] D. A. Turner and K. W. Ross. A lightweight currency paradigm for the P2P resource market. In *7th International Conference on Electronic Commerce Research*, June 2004. 24, 25, 29, 114

[303] Y. Vardi. Network tomography: Estimating source-destination traffic intensities from link data. *Journal of the American Statistical Association*, 91(433):365–377, 1996. 145

[304] H. Varian. Mechanism design for computerized agents. In *Proc. USENIX Workshop on Electronic Commerce*, New York, July 1995. 2, 4, 26, 154

[305] H. R. Varian. Equity, envy and efficiency. Working papers 115, Massachusetts Institute of Technology (MIT), Department of Economics, Aug. 1973. 27

[306] W. Vickrey. Counterspeculation, auctions, and competitive sealed tenders. *The Journal of Finance*, 16(1):8–37, 1961. 27, 48

[307] V. Vishnumurthy, S. Chandrakumar, and E. G. Sirer. KARMA: A secure economic framework for P2P resource sharing. In *Proc. of Workshop on the Economics of Peer-to-Peer Systems*, 2003. 8, 23, 41, 114

[308] B. D. Vleeschauwer, F. D. Turck, B. Dhoedt, and P. Demeester. On the construction of QoS enabled overlay networks. *Quality of Service in the Emerging Networking*, pages 164–173, September 2004. 112

[309] C. A. Waldspurger, T. Hogg, B. A. Huberman, J. O. Kephart, and W. S. Stornetta. Spawn: A distributed computational economy. *IEEE Trans. Software Eng.*, 18(2):103–117, 1992. 27, 74, 114

[310] B. Wang, J. Kurose, P. Shenoy, and D. Towsley. Multimedia streaming via TCP: an analytic performance study. In *MULTIMEDIA '04: Proceedings of the 12th annual ACM international conference on Multimedia*, pages 908–915, New York, NY, USA, 2004. ACM. 117

[311] F. Wang, J. Liu, and Y. Xiong. Stable peers: Existence, importance, and application in peer-to-peer live video streaming. In *INFOCOM*, Phoenix, USA, April 2008. 23

[312] J. H. Wang, D. M. Chiu, and J. C. Lui. A game-theoretic analysis of the implications of overlay network traffic on ISP peering. *Computer Networks*, 52(15):2961 – 2974, 2008. Complex Computer and Communication Networks. 9, 49

[313] W. Wang and B. Li. Market-driven bandwidth allocation in selfish overlay networks. In *INFOCOM 2005, 24th Annual Joint Conference of the IEEE Computer and Communications Societies*, pages 2578–2589. IEEE, 2005. 24, 114

[314] J. G. Wardrop. Some theoretical aspects of road traffic research. In *Proceedings of the Institute of Civil Engineers, Pt II*, volume 1, pages 325–378, 1952. 48

[315] D. J. Watts. *Small Worlds – The Dynamics of Networks between Order and Randomneess*. Princeton University Press, Princeton, New Jersey, 1999. 164

[316] B. Wilcox-O'Hearn. Experiences deploying a large-scale emergent network. In *IPTPS '01: Revised Papers from the First International Workshop on Peer-to-Peer Systems*, pages 104–110, London, UK, 2002. Springer-Verlag. 22, 23, 45, 164

[317] O. E. Williamson. *The Economic Institutions of Capitalism*. Free Press, December 1985. 155

[318] D. Wu, C. L. Y. Liu, and K. Ross. View-upload decoupling: A redesign of multi-channel P2P video systems. In *Proceedings of IEEE Conference on Computer and Communications (INFOCOM Mini-Conference '09)*, 2009. 69

[319] D. Wu, Y. Liu, and K. Ross. Queuing network models for multi-channel P2P live streaming systems. In *Proceedings of IEEE Conference on Computer and Communications (INFOCOM '09)*, 2009. 69

[320] H. Xie, Y. R. Yang, A. Krishnamurthy, Y. G. Liu, and A. Silberschatz. P4P: Provider portal for applications. *SIGCOMM Comput. Commun. Rev.*, 38(4):351–362, 2008. 9, 126, 127, 151

[321] B. Yang and H. Garcia-Molina. PPay: micropayments for peer-to-peer systems. In *CCS '03: Proceedings of the 10th ACM conference on Computer and communications security*, pages 300–310, New York, NY, USA, 2003. ACM Press. 24, 29

[322] X. Yang. Auction, but don't block. In *NetEcon '08: Proceedings of the 3rd international workshop on Economics of networked systems*, pages 19–24, New York, NY, USA, 2008. ACM. 9, 49

[323] H. Yu, M. Kaminsky, P. B. Gibbons, and A. Flaxman. SybilGuard: defending against sybil attacks via social networks. In *Proceedings of SIGCOMM '06*, pages 267–278, New York, NY, USA, 2006. ACM Press. ISBN 1-59593-308-5. doi: http://doi.acm.org/10.1145/1159913.1159945. 3, 54, 78

[324] E. W. Zegura, M. H. Ammar, Z. Fei, and S. Bhattacharjee. Application-layer anycasting: a server selection architecture and use in a replicated Web service. *IEEE-ACM Trans. Netw.*, 8(4):455–466, 2000. URL `citeseer.ist.psu.edu/article/zegura00applicationlayer.html`. 20

[325] J. Zhang, L. Liu, L. Ramaswamy, and C. Pu. Peercast: Churn-resilient end system multicast on heterogeneous overlay networks. *J. Netw. Comput. Appl.*, 31(4):821–850, 2008. 16

[326] M. Zhang, L. Zhao, Y. Tang, J.-G. Luo, and S.-Q. Yang. Large-scale live media streaming over peer-to-peer networks through global internet. In *P2PMMS'05: Proceedings of the ACM workshop on Advances in peer-to-peer multimedia streaming*, pages 21–28, New York, NY, USA, 2005. ACM. ISBN 1-59593-248-8. 16, 19

[327] X. Zhang, J. Liu, B. Li, and Y.-S. Yum. CoolStreaming/DONet: a data-driven overlay network for peer-to-peer live media streaming. *INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE*, 3:2102–2111 vol. 3, 13-17 March 2005. 16, 19, 117

[328] Z. Zhang, S. Chen, and M. Yoon. MARCH: A distributed incentive scheme for peer-to-peer networks. In *Proceedings of INFOCOM 2007*, pages 1091–1099, May 2007. 8, 69, 114

[329] B. Y. Zhao, J. D. Kubiatowicz, and A. D. Joseph. Tapestry: An infrastructure for fault-tolerant wide-area location and routing. Technical Report UCB/CSD-01-1141, EECS Department, University of California, Berkeley, Apr 2001. 17, 20

[330] Q. Zhao, J. Zhang, and J. Xu. Combating hidden action in unstructured peer-to-peer systems. *Communications and Networking in China, 2006. ChinaCom '06. First International Conference on*, pages 1–5, 25-27 Oct. 2006. 43, 149

[331] H. Zhu, B. Smith, and T. Yang. Hierarchical resource management for web server clusters with dynamic content. In *SIGMETRICS '99: Proceedings of the 1999 ACM SIGMETRICS international conference on Measurement and modeling of computer systems*, pages 198–199, New York, NY, USA, 1999. ACM Press. 15