Distributed Resource Management with Monetary Incentives

vorgelegt von
Diplom-Mathematiker
Andreas Tanner
aus Potsdam

von der Fakultät IV – Elektrotechnik und Informatik –
der Technischen Universität Berlin
zur Erlangung des akademischen Grades
Doktor der Naturwissenschaften
Dr. rer. nat.
genehmigte Dissertation

Promotionsausschuß:

Vorsitzender:	Prof. Dr.-Ing. Adam Wolisz
Betreuer:	Prof. Dr. Hans-Ulrich Heiß
Gutachter:	Prof. Dr. Kurt Geihs
Gutachter:	Prof. Giuseppe Persiano, Ph.D.

Tag der wissenschaftlichen Aussprache: 7. Juli 2005

Berlin 2005
D 83

Zusammenfassung

Moderne Kommunikationsnetzwerke erlauben die gleichzeitige Ausführung einer Vielzahl von Anwendungen. So ist es nicht verwunderlich, dass Mechanismen zur Ressourcenvergabe, seien es Übertragungsbandbreite oder Rechenkapazität, in der Informatik seit langer Zeit erforscht werden. Jedoch gehen die klassischen Mechanismen von *kooperierenden Nutzern*, die nicht versuchen, das System zu ihrem Vorteil zu manipulieren, aus. Andererseits wird die informationstechnische Infrastruktur schon aufgrund der hohen Installationskosten von Nutzern ohne gemeinsame Interessen und über Firmengrenzen hinweg genutzt. Stehen nun Ressourcen in begrenzter Menge zur Verfügung, ist die Entstehung von Allokationskonflikten unvermeidlich. Ein natürlicher Ansatz zur Lösung solcher Konflikte ist die Etablierung eines *Marktes* für die Ressourcenvergabe.

Diese Arbeit schlägt Marktmechanismen für die Ressourcenvergabe in verteilten Computersystemen vor. Wir präsentieren ein neues, budgetausgeglichenes Preisbildungsschema für kombinatorische Tauschmärkte, welches die Berechnung der Akzeptanz von Geboten unabhängig agierender Handelspartner erlaubt. Ebenso haben wir eine neue Methode zur Synchronisierung der Gebote entwickelt und zeigen, dass sie einer periodischen oder zufälligen Marktbereinigung überlegen ist. Die neu entwickelten Mechanismen wenden wir auf die Bandbreitenvergabe für Punkt-zu-Punkt-Kommunikation an und zeigen mittels einer Simulationsrechnung, dass die Auktionierung der Bandbreite für einen großen Teil des Parameterraumes zu höherer Effizienz als ein Fixpreisverkauf führt, obwohl die Auktionierung – im Unterschied zur Wahl eines optimalen Fixpreises – keine Informationen über die statistische Verteilung der Gebote der Nutzer benötigt.

Die Situation für Gruppenkommunikation erweist sich als schwieriger. Kellys klassischer Equlibriums-Mechanismus für die Bandbreitenvergabe verliert bei der Anwendung auf Gruppenkommunikation seine Effizienz. Wir präsentieren eine – nicht budgetausgeglichene – Verallgemeinerung von Feigenbaums Grenzkostenmechanismus auf ein Gruppenkommunikationsszenario mit Publish/Subscribe-Struktur, und entwickeln einen Algorithmus zur effizienten, verteilten Preisberechnung.

Abstract

Modern communication networks handle millions of applications simultaneously, and mechanisms of resource sharing, most prominently sharing of data transmission bandwidth and processing power, between competing jobs have been considered in computer science since its beginning. Classical mechanisms, however, balance claims of *cooperating* users that do not try to manipulate the system to their advantage. Due to their cost of installation, information technology infrastructure has to be used by clients with no joint interest: by individual users and accross borders of companies. With resources available only in a limited quantity, allocation conflicts do arise. It is natural to apply the classical remedy for conflict resolution and install a *market* for the system's resources.

This thesis proposes market mechanisms for resource allocation in distributed computer systems. We define a new budget-balanced pricing scheme for combinatorial exchanges that allows matching of bids of autonomous buyers and sellers. We suggets a new bid synchronization rule and prove that it performs superior to periodic and random bid clearing. We give an application of a combinatorial exchange to unicast bandwidth allocation. We demonstrate by a simulation that, for a large part of the parameter space, auctioning bandwidth performs superior to fixed price bandwidth sale, while not requiring prior information on the distribution of bids.

The situation for unicast cost sharing is more complicated. After proving that the classical equilibrium mechanism of Kelly can't looses much efficiency if applied to group communication, we present a generalization of Feigenbaum's adoption of marginal cost pricing to publish/subscribe settings. We also develop an algorithm for efficient distributed price computation.

Danksagung

Diese Arbeit entstand während meiner Tätigkeit als wissenschaftlicher Mitarbeiter am *Institut für Intelligente Netze und Management Verteilter Systeme* an der TU Berlin.

Ich habe Herrn Prof. Dr. Kurt Geihs zu danken, der in seiner Zeit an der TU Berlin für eine stimulierende Arbeitsumgebung sorgte, die mir eigenständige Forschung ermöglicht hat, und der mich in der Wahl meines Forschungsthemas bestärkte.

Herr Prof. Dr. Hans-Ulrich Heiss stellte sich freundlicherweise als Gutachter zur Verfügung und gab mir wertvolle Kritik und Anregungen in der Phase der Fertigstellung der Arbeit.

Gero Mühl ermutigte mich nachdrücklich, meine Ergebnisse zu Papier zu bringen. Er und Michael A. Jaeger waren außerordentlich hilfreiche Koautoren und haben mich durch unbeirrtes Insistieren zu größerer Präzision gezwungen. All meinen Kollegen bin ich für die vielen anregenden Diskussionen, die mir halfen, die praktische Anwendbarkeit meiner Ideen nicht aus den Augen zu verlieren, zutiefst zu Dank verpflichtet.

Meiner Familie, und besonders meiner lieben Frau Ximena, danke ich für die Ermutigung, sich auf das Wagnis Forschung einzulassen, und für die Geduld und Nachsicht in der langen Zeit bis zur Fertigstellung der Arbeit.

Contents

1 Introduction

This thesis is about resource allocation in distributed systems by applying techniques from applied game theory and economic theory. While resource management has experienced extensive treatment since distributed systems have gained popularity, the economic perspective is still a non-traditional one. Traditional resource management and resource management with economic incentives have a common goal: efficient use of the resources - typically memory, computational power and, probably most importantly, bandwidth consumption for data transfer - in a distributed system.

But what is *efficiency*? Classical resource management has a simple answer to that: given some set of tasks, efficiency means to solve them as quickly and accurately as possible, preferably while consuming little resources. The tasks are assumed to be known, sometimes deterministically, sometimes they are assumed to be drawn from some random distribution of possible tasks. A prominent example of resource management in that sense is *task scheduling*.

It is a natural approach to build this negotiation on monetary incentives, and precisely this is what our research is about.

Microeconomic theory describes economic interactions between self-interested individuals:

> A distinctive feature of microeconomic theory is that it aims to model economic activity as an interaction of individual economic agents pursuing their private interests. [48, p.3]

Mechanism design develops market mechanisms like pricing and allocation rules that produce optimal outcomes if market participants are self-interested and there is incomplete information. Mechanism design has been applied to analyze traditional auction markets, as well as to suggest new pricing and allocation rules, for instance for auctioning frequency spectrums for mobile communication and broadcast and for electrical power markets.

We raise the following questions:

- Computers are not humans. Economic theory is, in large parts, an empiric science about behaviour of humans. How can one transfer results from economic theory to a setting where only computers interact?
- Moreover, even among economists, it is not undisputed under which circumstances market equilibria produce an efficient outcome. Is *economization* of

infrastructural resources actually desirable?

- Even if computer scientists are per se interested and involved in modern communication technologies, it is nevertheless a plausible questions what they can contribute to a theory developed by economists and mathematicians.

The purpose of this introduction is to discuss these points and to formulate *theses* that we claim to be proved by the remaining chapters of this work.

1.1 Applicability of economic theory

In a modern distributed system with an open architecture, the set of tasks to be done is to be negotiated. Similarly, there has to be an agreement about the consumable resources. The economist Paul Samuelson[1] defines economic science as follows:

> Economic is the study of how men and society end up *choosing*, with or without the use of money, to employ *scarce* productive resources that could have alternative uses, to produce various commodities and distribute them for consumption, now or in the future, among various people and groups in society. It analyzes the costs and benefits of improcing patterns of resource allocation.[65, p.3]

Note that our resource management setup shows almost all characteristics Samuelson associates with economic science: only *scarce* resources need management, and management essentially takes place by some kind of task priorization: if there is only one unsplittable task at a time, there is no point in resource management.

Samuelson states that economy studies behaviour of “men and society”. It is therefore an *empirical* science, and the truth of an economic theory is lastly measured by its consistency with observation. This is in clear contrast with computer science which develops techniques to program a computer. A computer program (we here understand this term in a very general sense, including for example communication protocols) is not “observed” in its behaviour. Rather, computer science aims to understand the “behaviour”, that is, the execution, of the program completely. Even if computer scientists sometimes also use *simulations* to analyze a program execution in some particular context, results that solely depend on observed behaviour in simulations are generally judged as not as satisfactory compared with results that are proven with mathematical rigour.

This rigour does hold, however, for Microeconomics and in particular, Mechanism design. The *private interests* mentioned above are understood as the *maximization of individual, real-valued utility*, where the maximization is a mathematical optimization problem with constraints and incomplete information on the side of the agents

[1] Nobel laureate of 1970

as well as the control instance. The "approximation" of homo sapiens by *homo oeconomicus* is the conditio sine qua non of microeconomics which is thus embedded in the *neoclassical* branch of economic theory. Neoclassics and the assumptions it is built upon have been, and still are, fiercly disputed among economists.

Here are a couple of objections posed by the criticists[2]

- The theory has a lot of paradoxical results. For instance, the classical theory predicts (Bertrand's paradoxon, see [81, p.116]) that under perfect competition, producers will sell their commodities at marginal cost. Clearly fixed costs are thus not covered, and consequently, all producers run losses.

- More generally, there are many situation where at equilibrium, no market participant earns a profit. One might argue that this yields a contradiction for a theory which is built on the very assumption that everybody should maximize his surplus.

- What are the implications of negative results? Many theorems, like the theorems of Gibbard-Satterthwaite and of Green-Laffont[3], prove the non-existence of mechanisms with desirable properties. The theory says little about what happens if these properties are relaxed.

- The most obvious argument against utility-maximization is probably that it is questionable how a consumer would quantify his utility, say, for watching a movie in a certain quality. The theory relies on that input variable, not distinguishing between "true utility" (the *gain* of the consumer in comparison with non-purchase, measured in money), and *substitutional value*, that is, the value the consumer assigns to the commodity, given the possibilities of alternative purchases. While the monetary gain is simply non-measurable for most consumer goods[4] , the substitutional value is problematic since alternatives may have a different cost structure, and long-term effects of switching commodities may be hard to anticipate.

- In a broader sense, the assumption of human *rationality* has been very much under discussion. This discussion was opened by Simon[5] [72].

[2]M. Burchardt [13] gives a (partly a little outdated) critical review of microeconomic theory. Even if he omits game theory and mechanism design, he discusses most of the following points.

[3]See the next chapter for details.

[4]or may be unknown in advance, as Simon [72, p.113] remarks:

> The consequences that the organism experiences may change its pay-off function – it doesn't know how well it likes cheese until it has eaten cheese.

We may add: Similarly, a computer user may not know which quality of service is necessary prior to using that service.

[5]Simon was awarded ACM's Turing award in 1975, and the Nobel price for economics in 1978.

> Because of the psychological limits if the organism (particularly with respect to computational and predictive ability), actual human rationality-striving can at best be an extremely crude and simplified approximation to the kind of global rationality that is implied, for example, by game-theoretic models. While the approximation that the organism employ may not be the best – even at the levels of computational complexity they are able to handle – it is probable that at great deal can be learnt about possible mechanisms from an examination of the schemes of approximation that are actually employed by human or other organisms.[72, p.101]

In detail, Simon considers the following obstacles that prevent humans from using a "globally rational" decision strategy:

1. Partially ordered utilities: Simon suggests that human perception of payoffs is represented by *vector functions* better than by scalars: because preferences of different people involved in a decision may be contradictive, because an individual may have more than one concern, and because there is uncertainty about the *possible consequences* of a decision.
2. Limited "computing power": Against the proposition of uncertain dynamics, "classicists" may hold that this can be modelled with probabilities. Simons replies are twofold: first, the humans have no knowledge about the applicable probability functions. Second, humans are incapable[6] of actually performing the required calculations for computing an optimal expected outcome. *Probability theory is not a substitute for missing knowledge,* and the fact that the distribution of some variable is unknown does not per se justify the claim that it is randomly distributed.[7] The argument of limited computing power has been intensively addressed by computer science's contributions to mechanism design. We will discuss details in the next chapter.

[6] We add: or sceptical on the relevance of the computation's output

[7] We do note that we are in contradiction with Laplace [44] here. For this outspoken believer in determinism, randomness is in all cases just a consequence of lacking knowledge rather than an objective state. Given two possible outcomes *success* and *failure* of some variable that has been observed n times before, the *Laplace principle (rule of succession)* stipulates that the probability that *success* occurs at the $n+1$st instance is $P($*success* at $n+1) = \frac{s+1}{n+2}$ where s is the number of successes observed during the first n observations. In particular, with no prior observation, the probability of *success* is $\frac{1}{2}$. More generally, Laplace suggests that given that no other information is available, all possible alternatives should be assumed to be of equal probability, that is, a uniform distribution should be assumed. We raise three points challenging this principle: First, it is an a priori assumption and can't be proven. Second, to be applicable, there must be a unique decomposition of the set of all states into elementary alternatives. Third, it does not help at all in the case of continuous alternatives within an unknown range.

> Simon introduces a new model of human decision making which has later been associated with the terms of *bounded*[8] *rationality* or *satisficing*. He suggests that rather than trying to maximize utility, humans set for themselves an *aspiration level*[9] which may change over time. Alternatives that meat this aspiration level are considered equally valued in the corresponding category, and other categories are used to define a preference.[10]

Cyert and March [17], see [70, p.469f] propose a model for the behavioural dynamics of managers, owners, employees, customers and creditors of firms which is based on Simon's satisficing. According to them, there is a complex interplay of the interests of the different parties. In particular, while classic theory assumes that firms follow a profit-maximizing strategies, Cyert and March suggest that the company managers (which, after all, implement the company's strategies) try to *satisfice* goals set to them by the owners while otherwise being mainly interested in the well-being of their organizatorial unit. Additional surplus generated in "good times" is buffered in *organisational slacks* which may be used for conflict-mediation when revenues fall.

---

[8] in [72, p.113], "limited"

[9] [72, p. 111]

[10] Interestingly, the concept of an "aspired profit" has been adopted in catholic economical ethics. John Paul II writes in *Centesimus annus* [36, par.35]:

> When a firm makes a profit, this means that productive factors have been properly employed and corresponding human needs have been duly satisfied. But profitability is not the only indicator of a firm's condition. It is possible for the financial accounts to be in order, and yet for the people Ů who make up the firm's most valuable asset Ů to be humiliated and their dignity offended. Besides being morally inadmissible, this will eventually have negative repercussions on the firm's economic efficiency. In fact, the purpose of a business firm is not simply to make a profit, but is to be found in its very existence as a community of persons who in various ways are endeavouring to satisfy their basic needs, and who form a particular group at the service of the whole of society. Profit is a regulator of the life of a business, but it is not the only one; other human and moral factors must also be considered which, in the long term, are at least equally important for the life of a business.

The pope's argument of *long term* consequences closely resembles Simon's *uncertainty* of the dynamics. However, the Pope argues *normatively* as opposed to Simon who suggests bounded rationality as a *descriptive model* of human decision-making.

Satisficing as a normative, of course, has been at all times a constant in philosophical-ethical thinking. Aristotle (Nicomachean Ethics, ch. 6-9 [6]) defines *eudaimonia* (perfect and complete happiness) as the state where a human assumes the highest virtues (most notably, contemplation), *given that material needs are met*. Aristotle emphasizes (par. 1179a) that eudaimonia, being a state of perfection, requires that these needs must be fulfilled to a degree such that additional commodities would not increase happiness.

1.2 Networks

The transformation of the *internet* that connected essentially academic, government or public institutions, to nowadays *world wide web* with millions of users, many of them private individuals, others members of companies as well as academic and other institutions, is a challenge to the designers of the communication protocols. Traditional protocol design tries to optimize under the assumption that all parties involved faithfully honour protocol intentions. Clearly, this can't be taken for granted in open systems where cooperation competes with self-interested action, and the idea of introducing monetary incentives is a compelling one from the first thought.

In this section, we will outline three approaches that analyze the interconnection of information networks and economics. First, we adopt the narrow view of the computer scientists incorporating monetary payments into network protocols in order to give self-interested clients incentives to coordinate their demand and use the network efficiently. A second line of research is subsumized under the term of *network industries* used by economists to describe industries with *network effects*. Telecommunication companies and internet service providers are typical examples.

The "computer scientist's" research line confines to "classical" strategyproof (or weaker) mechanisms. However, the last decade has seen a lot of development of game-theory based *auction theory*. Inspiration for most of the theory comes from government-run spectrum auctions, and some from electronic markets like ebay or electrical power markets. It is interesting to ask which of the "modern" results are relevant for the protocol designer. The third subsection gives a – very biased – overview of modern auction theory.

1.2.1 Computer networks as markets

As computer scientist looking through the economist's glasses, we will interpret network resources as scarce *commodities* and network clients as utility-maximizing *agents*. We then can directly apply microeconomic theory to model the dynamics of the system. The most prominent examples for this paradigm of thinking concern bandwidth allocation. Nisan and Ronen [57] give a simple model of *shortest path routing* that allows application of Vickrey-Grove-Clarke mechanism. In their model, the network is modelled by a directed graph G whose *nodes* represent the self-interested agents with whom their is associated a privately known *cost of routing a package*. The standard Vickrey-Grove-Clarke mechanism[11] yields a cost-covering, incentive-compatible mechanism that always finds the shortest (measured in costs) path between any two nodes in the network.

There are many refinements of that model, including ones that consider congestion costs, multicast and multiple service levels.

[11] see next chapter for details

The markets described in these papers are closed subsystems of the economy. There are no external effects, no competition from outside, no dynamics of cost structures or customer behaviour and no roles in the economy besides “consumer” and “producers” of resources. The market rules are “axioms” that, in the best case, are closely modelled along with “reality”; however, there is no interest in empirical confirmation of the results. Since the markets are formally and rigorously described, results have intrinsic value independent from empirical evidence.

Of course, the authors of these papers are well aware of the fact that their mechanisms are little used in practice, and do occasionally offer some speculations on why this is so:

> Our approach of using an existing network[12] protocol as a substrate for realistic distributed computations may prove useful generally in Internet-algorithm design, not only in routing or pricing problems. Algorithm design for the internet has the extra subtlety that adoption is not a decision by a systems manager, concerned only with performance and efficiency, but rather a careful compromise by a web of autonomous entities, each with its own interests and legacies.[23]

In [23], authors model the network and the cost structure as a graph. They then prove that a certain Vickrey-Groves-Clarke pricing scheme has nice properties (most notably, strategyproofness), and is unique with that property. They propose distributed algorithms for payment computations and analyze its complexity. They note that having a strategyproof pricing scheme (which is useful if peers possibly try to manipulate), and letting peers compute payments may be problematic and formulate an open problem addressing that issue. They also note that the Vickrey mechanism has a problem of overcharging in comparison with the actual costs, and, for a rare example, offer an argument based on empiric observation of real internet providers, that states that for the observed graph, Vickrey pricing would not lead to extensive overcharging.[13]

Even if the model could be extended to cover these points, the question whether it would be advisable for a service provider to adopt the mechanism wouldn’t be answered. Allowing interdomain routing means opening a new market, and the model can’t in principle foretell which consequences thus arise:

- Erosion of prices (if total traffic does not grow, but competition between AS increases), or

[12] The authors present a pricing scheme for interdomain routing that can be embedded in the existing Border Gateway protocol.

[13] We note that there are some more directions the model could be extended: the paper does not model capacity constraints, it rather assumes that payments grow linearly with the traffic ad infinitum. Moreover, it is assumed that all packages travelling between a fixed pair of sender and receiver, take identical routes.

- an increase in efficiency which could trigger a larger demand?
- Increased fixed costs or need for investing into new hardware, if traffic does increase significantly, or
- a larger revenue with only little marginal costs?

It is obvious that these questions cannot be answered by methods of mechanism design only. In a similar fashion, even if participation in the model is precipated, the model offers no guidance on how to compute the utilities given as bids. Rather, to derive statements on these points, the model needs to be embedded into a complete model of the economy, or at least a sufficiently closed subset of it. But this means that a discussion of the foundations of economy, as outlined above, can't be avoided.

1.2.2 Network industries

The economist interested in information technology will find the computer scientist's perspective far to "technology-centred". He would prefer establishing a market for commodities that make use of resources, rather than trying to price the resources themselves. Instead of deriving market rules from the technical system, he would ask his "technology experts" to implement market rules of his choice. The economist comes with his rich tradition of market analysis, and will gladly apply what he has discovered about markets for bread or railway tickets or oil to new "products" like internet access or mobile communication.

The economist, however, will notice that a theory that works well with markets for bread won't necessarily work for mobile communication. Oz Shy defines four main characteristics that distinguish markets for *network products* from classical markets [71, p.1]:

- *complementarity, compatibility and standards:* Network products are not used standalone. Computer hardware and software can be purchased independently and are produced by different manufacturers, but only together they are useful to the consumer. Manufacturers have to meet strategic decisions about which products they design to cooperate, be it hardware architectures, operating systems and applications, or mobile phones and wireless networks. If compatibility is chosen, firms, even competing ones, must find a modus vivendi to develop standards that enable interoperability.
- *consumption externalities:* Utility of network product greatly depends on the size of the network. Trivial examples are communication devices like fax machines that are useful only if they enjoy some degree of popularity.
- *switching costs:* Network products are complex. While it is easy to substitute corn for wheat, it is not easy for a company to switch the operating system of all their computers.

- *“Significant economies of scale in production”*, that is, production costs are highly non-linear in the production amount. Typically, marginal cost – like “producing” one more software licence, or routing one more IP package – are low in comparison to the “fixed” costs, say, of developing a new application, or setting up a new telephone line.

Clearly, the picture the economist looks at is much more complex. He is deeply involved in the disputes that concern economic behaviour of humans. Although we are not aware of any example, we maintain that it is quit possible to develop an economic theory of network industries that builds on individuals that are not utility-maximizing.

Let us consider how the AS interdomain routing problem described in the previous section looks from a “network industry” point of view. Admitting interdomain transit traffic adds complexity to the market of data routing. In addition to offering a service to end consumers, a provider can now try to establish itself as a backbone transit “hub” that routes traffic from other providers. From an economic point of view, this is equivalent to *outsourcing* the traffic routing from the service offered to consumers, and can be compared with a railway company renting its tracks to other carriers. Adopting the extended business plan requires an extension of the Border Gateway protocol (BGP). Clearly, the amount and structure of competition will depend on mutual compatibility, and establishing one or multiple industry standards needs a lot of strategic consideration. Marginal routing costs are low in comparison with the cost of the infrastructure and the costs of changing an established protocol.

1.2.3 Auction theory

There has been extensive work in auction theory in the last fifteen years. Most of the game-theoretic analysis was inspired by the need for efficient allocation of spectrum licenses for broadcast and telecommunication. While auctions have traditionally been used by governments for property, eg land, sale and for procurement, the task of distributing these licenses pose a couple of new challenges: Most notably, the value of these licenses is difficult to estimate. On one hand, spectra are clearly a *scarce* resource with no production costs. In that sense, they compare with treasures of the soil. Explanation for market prices for treasures of the soil was a challenge to economists in the 18th and 19th century and lead to a refutation (at least partial) of the labour theory of value and the establishment of the marginal value theory. This theory estimates the per-unit price of a commodity to equal the additional utility generated if one more unit of the commodity is available, thus solving the “paradoxon” that while water has a higher utility than diamonds, the price for diamonds is nevertheless much higher.

In order to apply marginal value theory to spectrum licenses, one would need to forecast the revenue that companies can generate from the licenses. This, however, is difficult since governments don't know business plans of private firms. For some time, mobile phone licenses in the US were assigned by lottery, and it was hoped that an efficient allocation would evolve from secondary trade. The result was a fragmented and rather inefficient market, and US agencies became ready to adopt other schemes. *Auctions* were seen as a tool to take advantage of the competition between interested firms to extract a maximum of willingness to pay. Indeed auctions have performed, in some cases, very successful, while producing disappointing results in others.

The work of Vickrey on second price auctions [78] has served as inspiration for a tremendous amount of papers that generalized his results to many settings. Nevertheless, the few examples were pure second price auctions have been used for spectrum license sale have ended with extremely pure results. Vickrey auctions have a couple of disadvantages that make them unusable in some settings.

In 2004, two leading auction theorists, both involved in the design of spectrum and other government-run auctions in various countries, have published books [43, 50] on the modern economy of auctions. According to them, the following problems have to be addressed:

- Vickrey auctions, while stragegyproof, are not *groupwise* strategyproof: they are very sensitive to colluding bidders. Sellers can submit shill bids to increase prices. If the Vickrey mechanism is applied to combinatorial auctions, the resulting allocation is efficient but the payments can be low even if there are many high bids. Revenue may even shrink when more bids are submitted. Milgrom presents the following example[50, p.57f]:

  - Let there be two goods A and B, and four bidders b_1, b_2, b_3, b_4. Suppose that b_1 values the package of A and B with 10 and b_2 with 9. Suppose b_3 values A with 10 and b_4 values B with 10. b_1 and b_2 have no utility from a single item, while b_3 and b_4 have no additional utility from the second item.

    The efficient allocation, and thus the Vickrey-Grove-Clarke mechanism, gives A to b_3 and B to b_4. However, the Vickrey mechanism lets b_3 and b_4 pay nothing. If only b_1 and b_2 were present, b_1 would pay 9 for the package of A and B.

  In this example, the coalition consisting of the seller, b_1 and b_2 would prefer to trade among themselves. Milgrom [50, p.303] defines the *core* of an auction to be the set of all outcomes c with the property that there is no coalition which could find another outcome by trading only among themselves, such that all members of that coalition are better off with that outcome, than with c. Outcomes of Vickrey-Grove-Clarke mechanisms for combinatorial auctions are not generally in the core.

Even without this pathology of the Vickrey auction, bidders can collude by agreeing on some kind of "desirable" auction outcome. Klemperer [43, p.104] presents a couple of examples where competing bidder try to send signals through their bids to their competitors to persuade them to stop bidding on some items in exchange for others.

- Vickrey auctions are not even efficient if there is a potential of *mergers*[50, p.60]: in the above example, even if a merger between b_3 and b_4 would increase the valuation by a certain amount, say, 25%, the merged company would have to pay 10, thus suffering from a reduced total profit. In general, in the presence of complementary values, Vickrey auctions discourage mergers. The opposite is the case if goods are substitutes.

- *Attracting a sufficient number of bidders* is often more decisive for a successful auction than pricing rules. Klemperer and Milgrom quote the results of the New Zealand spectrum auctions as an example where, due to a large number of auctioned licenses for rather small areas, there were some licenses for which only one or very few bidders placed a bid at all. Since Vickrey pricing was used, some licenses went away for almost no payment, even if there was a single bid with a proper amount. The results lead to the demission of some of the politicians who could not advocate these results to the public, even if the question whether more revenue could have been generated with modified rules could not obviously be answered.

  Klemperer [43, p.42] formulates the *revenue equivalence theorem* which states that in auctions where every buyer wants to acquire at most one good, and buyer's types are independent private values, and supposed that bidders with the lowest possible type have zero gain from the auction, the seller's expected revenue does *only depend on the allocation rule*. In particular, an efficient auction would always make the seller either to retain the good, or to give it to the bidder with the highest valuation. The theorem states that the revenue is independent from the pricing rule. Consequently, in the case of a single item auction, the only way to make an auction efficient is to set an optimal *reserve price*, that is, the minimum bid that a seller would possibly accept. Indeed one can compute the optimal reserve price for various settings. However, Bulow and Klemperer have shown ([11], see [43, p.27]) that, under some weak assumptions, attracting a single additional bidder increases the expected seller's revenue more than setting an optimal reserve price ever could.

Klemperer[43, p.113] states:

> The fact that collusion and entry deterrence and, more generally, buyer market power is the key to auction problems suggests that auction design may not matter very much when there is a large number of potential

> bidders for whom entry to the auction is easy. For example, though much ink has been spilt on the subject of government security sales, auction design may not matter much for either price or efficiency in this case.

Continuing, Klemperer cautions that empirical literature on that topic, e.g. various analysis of US treasury auctions, is inconclusive, and irrelevance of auction design is not proven. We note that most theoretical results are proven only for single item auctions. In particular, there is no known revenue equivalence theorem for combinatorial auctions.

1.2.4 Characteristics of protocol design

This thesis proposes usage of monetary transfers as a tool for resource management whose intention is *an efficient usage of the available resources*. Economic theory is interesting to us as long as it says something about the behaviour of the system clients. This is a narrow focus: for instance, we do not pursuit the question whether it is advisable for the resource's owner to invest into producing additional resources. On the other hand, if a protocol would allow users to gain a better service, say, by forming a coalition with other users, then this would be relevant for the protocol designer.

The "market" of a communication network has a structure different from the market for spectrum licences or electric power generation. In the following, we will give some characteristics of the resource management market.

- *Large number of users.* Internet service providers typically have thousands of users accessing their network at any given time, and similar numbers hold for telecommunication providers. A news publishing service may easily have hundreds of subscribers.

- *Anonymity of usage.* Users may know some other network users, but do not generally know the resources they use at any given time. Due to the large number of users, they have little chance to know a significant portion of the usage profiles.

- *Interdependence of resources.* The level of service quality desired can only be provided by a *bundle* of resources. In a communication network, typically a chain of links is used. More generally, other resources besides bandwidth, like memory or computational power, have to be combined.

- *Impossibility of demand coordination.* In the case of the spectrum auctions, we had the phenomenon that some companies tried to coordinate demand in the sense that they proposed "splits" of the market, for example, by bidding aggressively for some bundle of licenses and leaving other bundles to competitors. We hold that this type of coordination is not possible for communication markets. There is no price differentiation in internet data traffic based on geographic realities, and it seems very improbable that users would accept one.

- *Large number of transactions.* A typical transaction for us is *sending some IP packages along a link*, or *connecting to some multicast stream for a duration of some seconds or minutes*. We expect that within an hour, thousands if not millions such transactions take place.

- *Automated bidding.* Bidding in this context will often be performed by automatized *agents*. Strategic bidding, therefore, is only feasible if it can be automatized, too. A consequence of that is, for example, that spontaneous "signalling" between agents is not possible.

- *Complexity is important.* Due to the large number of transactions, the amount of computation and communication required for placing bids, performing and communicating the matching is an important issue, particularly so because all of that has to take place in real time.

1.2.5 What are good protocols?

We are now ready to formulate the main results of this introduction: the criterions by whom we judge whether a given mechanism[14] is a good one.

- *Existence of dominant strategies or equilibria for single players.* Equilibria allow forecasting how the system state will develop. If collusion between players is not possible, it is safe to assume that players will follow dominant strategies if they exist. In view of the negative theoretical results, often there won't be good mechanisms with dominant strategies, and therefore, weaker equilibria like Bayes-Nash equilibria can be considered.

- *Efficiency at the equilibrium.* It would be desirable to have equilibria with maximized efficiency with respect to the accumulated utility. Often, selfish behaviour will lead to a suboptimal equilibrium.

- *Acceptable complexity (for users and owners) in the targeted usage scenario.* Combinatorial optimization problems are often NP-hard in the worst case. Avarage case complexity may be more encouraging, and acceptable approximations (though not with constant bound) may exist. A mechanism's complexity is acceptable if for the intended usage, the required optimization can be computed with sufficient quality.

[14]By mechanism, we mean here the part of a communication protocol that deals with monetary transfers.

- *Infeasibility of groupwise strategizing in the targeted usage scenario.* Even mechanisms with dominant strategies are not generally groupwise strategy-proof. The risks of collusion have to be analyzed with respect to the specific usage scenario.

In a subsequent step, it may be of interest how the protocol interacts with the current business model of the network owners. This, however, is not strictly a question for computer scientists and not in the focus of this work.

1.3 Related work

We give detailed account of related work in the relevant chapters. Here we mention only authors and works that opened major lines of research that we consider important for this thesis.

Distributed network resource sharing can be seen as a cooperation problem between selfish agents. Such problems were described first by Rosenschein and Zlotkin [86].

The paper of Nisan and Ronen [57] was the first to transfer mechanism design theory from microeconomic theory to computer science. The paper contains application scenarios for task scheduling and unicast end-to-end path-finding and triggered a huge amount of follow-up work. Nisan and Ronen's goal is the development of efficient mechanisms with dominant strategies. Nisan and Ronen's shortest-path scenario was developed further by Feigenbaum et al. in [57] and Hershberger et al. [32, 35]. Many settings involve the use of combinatorial auctions. In chapter 3, related work on combinatorial auctions and exchanges is presented.

Generalizations for multicast setting are treated in [54, 26, 25, 1, 7, 8]. The economical problem behind multicast settings is that of splitting the costs of a *public project* which is extensively treated in the literature, see [48] for a start. A general overview on applications of mechanism design with dominant strategies to computer science is given in [24].

Kelly et al. [42, 40, 41] focus on indirect mechanisms where network users control the flow via parameters that can be interpreted as monetary payment or payment in form of service degradation like increased latency. Their mechanisms don't have dominant strategies but often unique Nash equilibria which are approached in tatonnement processes. Roughgarden and Tardos [63] take a similar approach and analyze how network usage is affected by users that are selfish and sensible to latency.

1.4 Our work

This thesis tackles the problems posed above from different sides. After presenting relevant definitions and classic results and giving some general examples for applications in chapter 2, we motivate and develop a market type that is a specialized combinatorial exchange. We develop a new pricing scheme that satisfies budget-balance while preserving some of the useful properties of Vickrey-Grove-Clarke mechanisms. Furthermore we introduce a new clearing rule, the *commit window clearing*, and prove - empirically and partly analytically- its superiority to the well-known periodic and random clearing rules. In chapter 4, we give an application of the newly developed market to some unicast network resource managing scenarios with advanced reservations.

The following chapters consider multicast scenarios. Chapter 5 contains results of somewhat pessimistic nature: we show that mechanisms for multicast pricing implement generally quite inefficient equilibria: we show that the coordination ratio performs poorly for quite a couple of different utility functions. Finally, chapter 6 deals with publish/subscribe systems that we understand as special multicast systems. After giving a formal treatment that implements some message completeness guarantees, we develop a pricing mechanism with dominant strategies for these systems.

Thesis 1. *It is possible to implement a combinatorial exchange with budget-balanced pricing that guarantees sellers additional revenue compared to non-combinatorial markets.*

Thesis 2. *The efficiency of a combinatorial exchange market with autonomous traders depends on the used clearing policy. Commit window clearing generates a higher revenue compared with periodic and random clearing.*

Thesis 3. *Network bandwidth reservation with fixed reservation length can efficiently be built on a combinatorial exchange market that uses commit window clearing. Using a simple stochastic model that takes advantage of publicly known information on call characteristics, one can also implement reservations with unknown in advance length.*

Thesis 4. *If a network splits available (inelastic) supply in proportion with the user's willingness to pay, and users have linear utility, the allocation at the equilibrium has a coordination ratio of at least $\frac{3}{4}$, while for users with logarithmic utility functions, the coordination ratio is unbounded. In the corresponding multicast scenario, the coordination ratio is always unbounded.*

Thesis 5. *Marginal cost pricing is an efficient pricing for publish/subscribe scenarios if budget-balance is not required. Otherwise, the budget-balanced Shapley value pricing guarantees a minimal efficiency loss.*

2 Background: Game theory and mechanism design

This chapter presents definition and classical results from game theory, microeconomics, mechanism design and auction theory that is used in the later chapters. Main references are [48] for game theory, microeconomics and mechanism design, and [43, 50] for auction theory. We don't give proofs for well-known results, but do give detailed references.

2.1 Mechanisms

Let I a set of *players* or *agents* $i \in I$. Let X be a set of *alternatives*, or *outcomes*. Every $i \in I$ has a *utility profile* $u_i : X \mapsto \mathbb{R}$. Let $U_i \subseteq X^{\mathbb{R}}$ be the set of all possible strategy profiles for agent i.

For a vector $\vec{u} = (u_i : i \in I)$, let us write $\vec{u}|_{u_j = x}$ for the vector $(v_i : i \in I)$ with

$$v_j = \begin{cases} u_j & \text{if } j \neq i \\ x & \text{if } j = i \end{cases} \tag{2.1}$$

For a vector $\vec{u} = (u_j : j \in I)$, let $\vec{u}_{-i}$ denote $\vec{u}_{-i} = (u_j : i \neq j \in I)$.

Definition 1 (Mechanism)**.** *A* (direct) mechanism $\mathfrak{M}$ *is a tuple* $\mathfrak{M} = (o^{\mathfrak{M}}, p^{\mathfrak{M}})$ *such that*

- $o^{\mathfrak{M}}$ *is a* social choice *function that maps every* profile vector $\vec{u} = (u_i : i \in I)$ *to some outcome* $o^{\mathfrak{M}}(\vec{u}) \in X$*, and*

- $p^{\mathfrak{M}}$ *is a* payment function *mapping every* $\vec{u} = (u_i : i \in I)$ *to some* payment vector $p^{\mathfrak{M}}(\vec{u}) = (p_i^{\mathfrak{M}} : i \in I)$ *with* $p_i^{\mathfrak{M}} \in \mathbb{R}$*.*

$\mathfrak{M}$

- *is* deficit-free *if*
$$(\forall \vec{u}) \sum_{i \in I} p_i^{\mathfrak{M}}(\vec{u}) \geq 0, \tag{2.2}$$

- *is* budget-balanced *if*
$$(\forall \vec{u}) \sum_{i \in I} p_i^{\mathfrak{M}}(\vec{u}) = 0, \tag{2.3}$$

- *satisfies* voluntary participation *if*

$$(\forall \vec{u} = (u_i : i \in I))\,(\forall i)\, u_i\left(o^{\mathfrak{M}}(\vec{u})\right) - p_i^{\mathfrak{M}}(\vec{u}) \geq 0, \tag{2.4}$$

- *satisfies* consumer sovereignty *if for all* $i \in I$*, all* $x \in X$ *and all* $\vec{u} = (u_i : i \in I)$*, there is* $u_i' \in U_i$ *such that* $o^{\mathfrak{M}}\left(\vec{u}|_{u_i = u_i'}\right) = x$*.*

Example 2. *Consider the following scheduling problem:*

Let there be n *jobs and* m *processors such that* t_j^i *are processor* i*'s cost for processing job* j*. The vector* $(t_j^i : 1 \leq j \leq n)$ *is processor* i*'s* type*. Let* X*, the set of outcomes, be the set of all possible functions* $x : \{1, \ldots, n\} \mapsto \{1, \ldots, m\}$ *that assign to every job* j *(*$1 \leq j \leq n$*) a processor* i *(*$1 \leq i \leq m$*).*

Define $\mathfrak{M} = (o, p)$ *by*

$$o\left(t_j^i : 1 \leq i \leq m, 1 \leq j \leq n\right) \in \arg\min_{x \in X} \left\{ \sum_{i,j : x(j) = i} t_j^i \right\} \tag{2.5}$$

$$p\left(t_j^i : 1 \leq i \leq m, 1 \leq j \leq n\right) = \sum_{i,j : o(j) = i} t_j^i \tag{2.6}$$

Then $\mathfrak{M}$ *is a direct mechanism for* X *that is not deficit-free (since it makes only payments but does not generate any income). It does satisfy voluntary participation (since it compensates a processor for processing a job exactly with the amount of the claimed costs). It does not satisfy consumer sovereignty since a processor can't force to get a job assigned, even if he claims that he has zero costs of processing: there could be another processor with no costs either.*

2.1.1 Strategies

Definition 3 (Strategy)**.** *A* (pure) strategy *of agent* i *for mechanism* $\mathfrak{M}$ *is a mapping* $s : u_i \mapsto s(u_i)$ *from the set of utility profiles of* i *to itself.*

Definition 4 (Dominant strategy)**.** *A strategy* s *is* dominant *if for all profiles* $\vec{u} = (u_i : i \in I)$ *and for all* $u_i' \neq u_i$*,*

$$u_i\left(o^{\mathfrak{M}}(\vec{u}|_{u_i = u_i'})\right) - p_i^{\mathfrak{M}}\left(\vec{u}|_{u_i = u_i'}\right) \leq u_i\left(o^{\mathfrak{M}}(\vec{u})\right) - p_i^{\mathfrak{M}}(\vec{u}) \tag{2.7}$$

A strategy is strictly dominant*, if strict inequality holds in (2.7) for at least one profile vector* $\vec{u}$*.*

Definition 5 (Truthful mechanism)**.** *A mechanism* $\mathfrak{M}$ *is* (strictly) truthful *if for every agent* i*, the* truth-telling strategy*, that is, the strategy* $s(u_i) = u_i$*, is (strictly) dominant.*

Definition 6 (Implementable social choice function)**.** *A social choice function* o *is* implementable in dominant strategies *if there is a mechanism* $\mathfrak{M} = (o^{\mathfrak{M}}, p^{\mathfrak{M}})$ *such that there is a strategy vector* $\vec{s}$ *of dominant strategies in* $\mathfrak{M}$ *such that for all profiles* $\vec{u}$*,*

$$o(\vec{u}) = o^{\mathfrak{M}}(s_1(u_1), \ldots, s_n(u_n)) \tag{2.8}$$

In this case, we say that $\mathfrak{M}$ implements o*.*

Remark 7. *If* $\mathfrak{M} = (o^{\mathfrak{M}}, p^{\mathfrak{M}})$ *is truthful, then* $\mathfrak{M}$ *implements* $o^{\mathfrak{M}}$*.*

Example 8. *The mechanism in example 2 is* not *truthful. Agents are compensated with an amount equal to their claim. Therefore, increasing a claim such that the social choice function remains unchanged is more favourable to a processor than truth-telling.*

Note that this shows also that while the o *minimizes total cost based on the costs* claimed by the processors*, the social choice function that* $\mathfrak{M}$ *implements does not minimize social total costs (since the processors won't reveal their true costs).*

The notions of *dominant strategy* and *truthfulness* are strong ones: a dominant strategy has optimal performance, no matter which strategies are used by other players. The notion of *Nash equilibrium* is weaker: a stragegy vector is a Nash equilibrium if no single player can gain from unilateraly changing his strategy.

Definition 9 (Nash equilibrium)**.** *Let* $\vec{S} = (s_i : i \in I)$ *be a vector of strategies and let* $\vec{u} = (s(u_i) : i \in I)$*. We say that* $\vec{S}$ *is a* (pure) Nash equilibrium*, if for all* $i \in I$ *and* $u_i' \neq s(u_i)$*,*

$$u_i\left(o^{\mathfrak{M}}(\vec{u}|_{u_i=u_i'})\right) - p_i\left(o^{\mathfrak{M}}(\vec{u}|_{u_i=s(u_i)})\right) \leq u_i\left(o^{\mathfrak{M}}(\vec{u})\right) - p_i^{\mathfrak{M}}(\vec{u}) \tag{2.9}$$

We say that $\vec{S}$ *is a* strict Nash equilibrium *if strict inequality holds in (2.9).*

So if $\vec{S}$ is a Nash equilibrium, then no agent i has an incentive to divert from his strategy *provided that the others don't divert from theirs*.

2.1.2 Mixed strategies

Instead of following a deterministic strategy, an agent can *randomize* over his options.

Definition 10 (Mixed strategy)**.** *A* mixed strategy s *for agent* i *is a probability distribution over the set of all possible strategies of* i*.*

Let o *be a social choice function. For a vector* $\vec{s} = (s_i : i \in I)$ *of mixed strategies agents* $i \in I$*, we write* $u_i(o(\vec{s}))$ *and* $p_i(o(\vec{s}))$ *as a shortcut for the* expected value *of the random variables* $u_i(o(s^1, \ldots, s^n))$ *and* $p_i(o(s^1, \ldots, s^n))$*, where* s^i *are random variables with distribution according to* s_i*.*

The generalization of definitions 4 and 9 is straightforward:

Definition 11 (Dominant mixed strategy)**.** *A mixed strategy s is* dominant *if for all profiles $\vec{u} = (u_i : i \in I)$ and for all $u_i' \neq u_i$,*

$$u_i\left(o^{\mathfrak{M}}(\vec{u}|_{u_i=u_i'})\right) - p_i^{\mathfrak{M}}\left(\vec{u}|_{u_i=u_i'}\right) \leq u_i\left(o^{\mathfrak{M}}(\vec{u})\right) - p_i^{\mathfrak{M}}(\vec{u}) \tag{2.10}$$

with the expected value interpretation of definition 10. Similarly, a mixed strategy is strictly dominant*, if strict inequality holds in (2.7) for at least one profile vector $\vec{u}$.*

Definition 12 (Mixed Nash equilibrium)**.** *Let $\vec{S} = (s_i : i \in I)$ be a vector of strategies and let $\vec{u} = (s(u_i) : i \in I)$. We say that $\vec{S}$ is a* Nash equilibrium*, if for all $i \in I$ and $u_i' \neq s(u_i)$,*

$$u_i\left(o^{\mathfrak{M}}(\vec{u}|_{u_i=u_i'})\right) - p_i\left(o^{\mathfrak{M}}(\vec{u}|_{u_i=s(u_i)})\right) \leq u_i\left(o^{\mathfrak{M}}(\vec{u})\right) - p_i^{\mathfrak{M}}(\vec{u}), \tag{2.11}$$

again with the expected value interpretation of definition 10. We say that $\vec{S}$ is a strict Nash equilibrium *if strict inequality holds in (2.9).*

The following is obvious:

Fact 13. *If for $i \in I$, s_i is a dominant (mixed) strategy for agent i, then $(s_i : i \in I)$ is a (mixed) Nash equilibrium.*

2.1.3 Efficiency of equilibria

A "good" mechanism will maximize total welfare. This leads to the following definitions:

2.1.3.1 Dominant strategies

Definition 14 (Efficient social choice function)**.** *A social choice function o is* efficient*, if*

$$\sum_{i \in I} u_i\left(o(\vec{u})\right) \geq \sum_{i \in I} u_i\left(o'(\vec{u})\right) \tag{2.12}$$

for all social choice functions $o' \neq o$ and profile vectors $\vec{u} = (u_i : i \in I)$.

Definition 15 (Efficient mechanism)**.** *A mechanism $\mathfrak{M} = (o^{\mathfrak{M}}, p^{\mathfrak{M}})$ is* efficient *(in dominant strategies), if there is a strategy vector $(s_i : i \in I)$ such that for all user profile vectors $(u_i : i \in I)$,*

- *for every $i \in I$, s_i is a dominant strategy for i, and*

•
$$\sum_{i\in I} u_i \left(o^{\mathfrak{M}}(s_1(u_1),\ldots,s_n(u_n))\right) \geq \sum_{i\in I} u_i \left(o'(s_1(u_1),\ldots,s_n(u_n))\right) \tag{2.13}$$

for all o'.

$\mathfrak{M}$ *is* strictly efficient *if for all* $i \in I$, s_i *is strictly dominant.*

Note that

- although equation (2.13) does not depend on $p^{\mathfrak{M}}$, nevertheless $\mathfrak{M}$ being efficient *does* depend on $p^{\mathfrak{M}}$, since it depends on the payment function whether a given strategy is dominant, and
- it is neither necessary nor sufficient for $\mathfrak{M}$ being efficient that $o^{\mathfrak{M}}$ is efficient.

However, the following fact is a consequence of the *revelation principle* (see [48], proposition 23.C.1):

Fact 16. *If $\mathfrak{M}$ is efficient, then there is $\mathfrak{M}'$ such that $\mathfrak{M}'$ is equivalent to $\mathfrak{M}$ in the sense that for any vector $\vec{s}$ of dominant strategies in $\mathfrak{M}$, there is a vector $\vec{s}'$ of dominant strategies for $\mathfrak{M}'$ such that for any profile vector $\vec{u}$,*

$$o^{\mathfrak{M}}(s_1(u_1),\ldots,s_n(u_n)) = o^{\mathfrak{M}'}\left(s'_1(u_1),\ldots,s'_n(u_n)\right) \tag{2.14}$$

and

$$p^{\mathfrak{M}}(s_1(u_1),\ldots,s_n(u_n)) = p^{\mathfrak{M}'}\left(s'_1(u_1),\ldots,s'_n(u_n)\right) \tag{2.15}$$

2.1.3.2 Coordination ratio

It is safe to assume that "rational" (that is, utility maximizing) agents will play according to a dominant strategy equilibrium. Thus, a strictly efficient mechanism will yield an efficient outcome. This, however, is not generally true for Nash equilibria since games can have multiple Nash equilibria with different social surplus. Therefore, the definition of the coordination ratio contains a reference to a specific Nash equilibrium that can be dropped only if it is unique.

Definition 17 (Coordination ratio)**.** *Let $\vec{s}$ be a Nash equilibrium for $\mathfrak{M}$. The* coordination ratio *of $\vec{s}$ is defined to be*

$$r^{\mathfrak{M}}_{\vec{s}} = \inf_{\vec{u}=(u_i:i\in I)} \frac{\sum_{i\in I} u_i\left(o^{\mathfrak{M}}(s_1(u_1),\ldots,s_n(u_n))\right)}{\max_{x\in X}\sum_{i\in I} u_i(x)} \tag{2.16}$$

If $\vec{s}$ is the unique Nash equilibrium, we write $r^{\mathfrak{M}}$ for $r^{\mathfrak{M}}_{\vec{s}}$.

2.1.4 Existence of Nash equilibria

2.1.4.1 Existence of Nash equilibria for given mechanisms

Not every mechanism has a mixed Nash equilibrium:

Example 18. *Let $X = \{1, 2\}$ and let the profiles of agents 1 and 2 be given by*

$$u_1(1) = 1 \qquad u_1(2) = 0 \tag{2.17}$$
$$u_2(1) = 0 \qquad u_2(2) = 1 \tag{2.18}$$

Now let the social choice function o be defined by

$$o(u_1, u_2) = \begin{cases} 1 \text{ if } u_1(1) \geq u_2(1) \\ 2 \text{ if } u_1(1) < u_2(1) \end{cases} \tag{2.19}$$

and let the payment function p be the zero function. That means, "winner" in this game is the agent who reports the higher utility. Clearly, there is no Nash equilibrium for (o, p) (not even for a mixed strategy), since the agent that reported the smaller utility could always have won the game by reporting a higher one.

However, mixed Nash equilibria do exist under quite general assumptions:

Theorem 19 (see [48], proposition 8.D.3)**.** *Assume that U_i are compact and convex subspaces of some Euclidian space for $i \in I$, and the $u_i(u_1, \ldots, u_n) := u_i(o(u_1, \ldots, u_n))$ and $p_i(u_1, \ldots, u_n) := p_i(o(u_1, \ldots, u_n))$ are continuous in $(u_1, \ldots, u_n)$ and quasi-concave[1] in every u_i. Then there is a mixed strategy Nash equilibrium for the mechanism (o, p).*

A mixed Nash equilibrium does also exist if there are finitely many agents with finitely many strategies:

Theorem 20 (see [48], proposition 8.D.2)**.** *Let I be finite and suppose that for every $i \in I$, U_i is finite. Then any mechanism has a mixed Nash equilibrium.*

Example 21. *Suppose that there are users $1, \ldots, n$ of some link of capacity 1. Suppose that the link's bandwidth is split in proportion with the* bid amounts $b_1, \ldots, b_n$ *that the users attach to their bids. Users pay an amoutn on money equal to their bids. User i's surplus then is*

$$s_i = u_i \left(\frac{b_i}{\sum_{j=1}^{n} b_j} \right) - b_i \tag{2.20}$$

The above theorem implies that there is a Nash equilibrium for the associated game if all u_i are quasi-concave. Chapter 5 gives more results for this scenario.

[1] $f : \mathbb{R} \supseteq D \mapsto \mathbb{R}$ is *quasi-concave* if for all $y \in \mathbb{R}$, $\{x \in D : f(x) \geq y\}$ is convex.

Note that even in the finite case, pure Nash equilibria do not generally exist:

Example 22 (Matching pennies). *Let X and the utility profiles be as in example 18 and let o be given by*

$$o(u_1, u_2) = \begin{cases} 1 \text{ if } u_1(1) = u_2(1) \\ 2 \text{ if } u_1(1) \neq u_2(1) \end{cases} \tag{2.21}$$

and let p be the zero function. Then there is no pure Nash equilibrium for (o, p) (since the "looser" of the game would win if he (and only he) changed his value of $u_i(1)$). There is, however, a mixed Nash equilibrium that lets every player randomly choose between his two possible choices for $u_i(1)$. In fact, even if only one of the agents chooses $u_i(1)$ randomly and the other agent follows any strategy, the yielded strategy set is a mixed Nash equilibrium. This shows that Nash equilibria are not unique in general.

2.1.4.2 Bayesian Nash equilibria and implementable choice functions

Definition 9 of Nash equilibrium required that for every i with type u_i and *any* vector of the "remaining" types $\vec{u}_{-i}$, agent i is better off playing according to his equilibrium strategy provided that the remaining agents do.

If we assume that the agent's types u_i are *random variables* drawn from U_i according to some statistical distribution, we can weaken the notion of Nash equilibria even further:

Definition 23 (Bayesian Nash equilibrium). *Suppose that $\vec{u} \in \times_{i \in I} U_i$ are drawn according to some probability distribution F. Let $\mathfrak{M} = (o^{\mathfrak{M}}, p^{\mathfrak{M}})$. A strategy vector $\vec{s}$ is a* Bayesian Nash equilibrium *if for all i and all $u_i, u_i' \in U_i$,*

$$E_{\vec{u}_{-i}}\left[u_i\left(o^{\mathfrak{M}}(u_i', \vec{u}_{-i})\right) - p_i\left(o^{\mathfrak{M}}(u_i', \vec{u}_{-i})\right)\right] \leq E_{\vec{u}_{-i}}\left[u_i\left(o^{\mathfrak{M}}(u_i, \vec{u}_{-i})\right) - p_i^{\mathfrak{M}}(u_i, \vec{u}_{-i})\right] \tag{2.22}$$

where the expected value E is taken over all possible $\vec{u}_{-i}$ subject to F conditioned on u_i.

Definition 24 (Expected externality mechanism). *Let o be a social choice function. The* expected externality mechanism *for o is the mechanism $\mathfrak{M} = (o, p)$, where for $\vec{u} = (u_i : i \in I)$*

$$p_i(\vec{u}) \;=\; -E_{\vec{v}'_{-i}}\left[\sum_{j \neq i} v_j\left(o(u_i, \vec{v}_{-i})\right)\right] + \left(\frac{1}{n-1}\right)\sum_{j \neq i}\left(E_{\vec{v}_{-i}}\left[\sum_{k \neq j} v_k(o(u_j, \vec{v}_{-j}))\right]\right) \tag{2.23}$$

The following is well-known (see [48, p.886f]):

Theorem 25. *If o is efficient and the agent's types are independently from each other, then the expected externality mechanism for o is budget balanced and implements o in Bayesian Nash equilibria.*

While the expected externality mechanism is efficient and budget balanced, it will not generally satisfy the condition of voluntary participation. The Myerson-Satterthwaite theorem ([56], see [48], proposition 23.E.1[2]) states that for a specific setting, there are no budget-balanced mechanisms that implement an efficient social choice function in Bayesian Nash equilibria with voluntary participation.

Theorem 26 (Myerson-Satterthwaite theorem)**.** *Let there be two agents 1,2 and let $X = \{1, 2\}$ and*

$$u_i(x) = \begin{cases} u_i \text{ if } x = i \\ 0 \text{ otherwise} \end{cases} \tag{2.24}$$

for $i = 1, 2$. Suppose that the u_i are independently drawn from intervals $[u_i^{min}, u_i^{max}]$ with strictly positive densities, and $(u_1^{min}, u_1^{max}) \cap (u_2^{min}, u_2^{max}) \neq \emptyset$. Then there is no efficient social choice function o that is implementable in Bayesian Nash equilibria with voluntary participation and budget-balanced payment rule.

2.1.5 Existence of dominant strategies

2.1.5.1 Choice functions that are implementable in dominant strategies

While the existence of mixed Nash equilibria is assured in many cases, far less mechanisms have dominant strategies for their participants. Roberts [61] (Theorem 3.1) has given a characterization of social choice functions that are implementable in dominant strategies:

Theorem 27. *Let o be a social choice function implemented by $\mathfrak{M}$ in dominant strategies. Assume that for all $x \in X$, there is $\vec{u}$ with $o(\vec{u}) = x$. Then there is a* weight vector $\vec{k} = (k_i : i \in I)$ *with $k_i \geq 0$ and some i with $k_i > 0$, and a function $F : X \mapsto \mathbb{R}$, such that for all $\vec{u}$,*

$$o(\vec{u}) \in \arg\max_{x \in X} \left\{ \sum_{i \in I} k_i \cdot u_i(x) \, + \, F(x) \right\} \tag{2.25}$$

If we take F as the utility function of some "additional" agent i_0, Theorem 27 can be interpreted as saying that exactly those choice functions are implementable in dominant strategies that maximize *weighted total surplus* for some weight vector $\vec{k}$.

[2] Note that the condition of budget-balance is not explicitly mentioned there but derived from the context.

2.1.5.2 Vickrey Groves Clarke mechanisms

Definition 28 (Vickrey-Groves-Clarke mechanism)**.** *A mechanism* $\mathfrak{M} = (o^{\mathfrak{M}}, p^{\mathfrak{M}})$ *is a* Vickrey-Groves-Clarke (VGC) mechanism *if*

1. $o^{\mathfrak{M}}$ *is efficient, and*
2. $p^{\mathfrak{M}}$ *has the form* $(p_i^{\mathfrak{M}} : i \in I)$ *with*

$$p_i^{\mathfrak{M}}(\vec{u}) = -\left(\sum_{j \neq i} u_j\left(o^{\mathfrak{M}}(\vec{u})\right)\right) + h_i(\vec{u}_{-i}) \tag{2.26}$$

for some function h_i *which does not depend on* u_i*.*

A classic result (see e.g. [48], proposition 23.C.4) is

Theorem 29. *If* $\mathfrak{M}$ *is a Vickrey-Groves-Clarke mechanism, then* $\mathfrak{M}$ *is truthful and efficient.*

Green and Laffont [30] proved that

Theorem 30. *If for every* $i \in I$ *and every function* $f : X \mapsto \mathbb{R}$*, there is* $u_i \in U_i$ *with* $u_i(x) = f(x)$ *for all* $x \in X$*, then every truthful efficient mechanism is a Vickrey-Groves-Clarke mechanism.*

2.1.5.3 Budget balance

VGC mechanism are in general not budget balanced. In fact, Green and Laffont [30] showed that they are never, under the prerequisites of Theorem 30:

Theorem 31. *If for every* $i \in I$ *and every function* $f : X \mapsto \mathbb{R}$*, there is* $u_i \in U_i$ *with* $u_i(x) = f(x)$ *for all* $x \in X$*, then there is no truthful efficient budget-balanced mechanism.*

2.2 Application of VGC mechanisms to allocation problems

Definition 32 (Allocation problem)**.** *An* allocation problem *for agents* $i \in I$ *and a set of goods* J *is a set of social choices* $X \subseteq \{(k_j^i : i \in I, j \in J, k_j^i \in \mathbb{R})\}$ *with a set of utility profiles* $(U_i : i \in I)$ *such that for all* i *and* $u_i \in U_i$*,*

$$\left(\forall_{\vec{k},\vec{k}' \in X}\right)\left[\left((\forall_{j \in J})\left(\vec{k}_j^{-i} = \vec{k}_j^{\prime -i}\right)\right) \Rightarrow \left(u_i(\vec{k}) = u_i(\vec{k}')\right)\right] \tag{2.27}$$

and for all agents i*,*

$$\left((\forall_j) k_j^i = 0\right) \Rightarrow u_i(k_j^{i'} : i' \in I, j \in J) = 0 \tag{2.28}$$

The set J *is called* set of goods*.*

k^i_j can be understood as the quantity of good j that is allocated to agent i. Equation (2.27) says that an agent is indifferent between two social choices if the quantity of goods awarded to him is identical in both choices. Equation (2.28) says that agents have zero utility if they don't get any good.

Example 33 (Task scheduling)**.** *Let there be agents* $1, \ldots, n$ *and a set of* tasks *each of whome can be processed by any of the agents. Suppose that processing task* j *by agent* i *induces cost* $c_{i,j}$ *to agent* i*. A function* $f : J \mapsto I$ *that assigns to every task* j *some agent* i *can be interpreted as an allocation function by setting* $k^i_j = 1$ *iff* $f(j) = i$ *and 0 otherwise. Agent* i*'s utility from function* f *then is* $u_i(f) = -\sum_{\{j \in J : f(j) = i\}} c_{i,j}$.

Example 34 (Combinatorial auction)**.** *Fix a set* I *of agents and a set* J *of goods. Define*

$$X = \left\{ \left(k^i_j : i \in I, j \in J \right), k^i_j \in \{0, 1\}, \sum_{i \in I} k^i_j \leq 1 \text{ for all } j \in J \right\} \tag{2.29}$$

This models a market where agents i *compete about goods* j *each available in exactly one copy. Retaining goods is allowed (it would not if we would require* $\sum_i k^i_j = 1$ *for* $j \in J$*).*

Equation (2.27) allows us to write $u_i(G)$ *as a shortcut for* $u_i(k^i_j : i \in I, j \in J)$ *where* $G = \{j \in J : k^i_j = 1\}$.

Let o *be a social choice function for* X*. For a profile vector* $\vec{u}$*, write*

$$o^i(\vec{u}) = \{j \in J : k^i_j = 1, (k^i_j : i \in I, j \in J) = o(\vec{u})\} \tag{2.30}$$

and

$$V^o(\vec{u}) = \sum_i u_i(o^i(\vec{u})) \tag{2.31}$$

So $o^i(\vec{u})$ *is the set of goods allocated to agent* i *by* o *if the utility profile is* $\vec{u}$*.*

Now let $\mathfrak{M} = (o^{\mathfrak{M}}, p^{\mathfrak{M}})$ *be a VGC mechanism for* X *that satisfies voluntary participation and the* no positive transfers condition $p^{\mathfrak{M}} \geq 0$*. This implies* $p^{\mathfrak{M}}_i(\vec{u}|_{u_i \equiv 0}) = 0$*. The efficiency condition (2.12) can now be written as*

$$o^{\mathfrak{M}}(\vec{u}) \in \arg_{o : \times_i U_i \mapsto X} \max \sum_i u_i(o^i(\vec{u})) \tag{2.32}$$

For the payment rule, we get according to (2.26) (and dropping the superscript $\mathfrak{M}$ *of* $o^{\mathfrak{M}}$ *for notational convenience)*

$$p^{\mathfrak{M}}_i(\vec{u}) = -\left(\sum_{i' \neq i} u_{i'} \left(o^{i'} (\vec{u}) \right) \right) + h_i \left(\vec{u}_{-i} \right) \tag{2.33}$$

$$= -V^o(\vec{u}) + u_i(o^i(\vec{u})) + h_i \left(\vec{u}_{-i} \right) \tag{2.34}$$

So

$$0 = p_i^{\mathfrak{M}}(\vec{u}|_{u_i \equiv 0}) = -V^o(\vec{u}|_{u_i \equiv 0}) + 0 + h_i\left(\vec{u}_{-i}\right) \tag{2.35}$$

and consequently

$$h_i\left(\vec{u}_{-i}\right) = V^o(\vec{u}|_{u_i \equiv 0}) \tag{2.36}$$

Finally we get

$$p_i^{\mathfrak{M}}(\vec{u}) = V^o(\vec{u}|_{u_i \equiv 0}) - V^o(\vec{u}) + u_i(o^i(\vec{u})) \tag{2.37}$$

The term $V^o(\vec{u}) - V^o(\vec{u}|_{u_i \equiv 0})$ *is called* Vickrey discount $\Delta^o_{\text{vic}}(\vec{u})$*. So*

$$p_i^{\mathfrak{M}}(\vec{u}) = u_i(o^i(\vec{u})) - \Delta^o_{\text{vic}}(\vec{u}) \tag{2.38}$$

The value of the Vickrey discount is exactly the marginal social surplus contributed by agent i. *Agent i pays his utility discounted by this contribution.*

Note that $\mathfrak{M}$ is not budget balanced. Rather, if the utilities that the agents draw from the goods are nonnegative, the mechanism generates a surplus, the "revenue" of the auction.

Example 35 (Combinatorial exchange)**.** *In a combinatorial auction, the seller is not represented by an agent. It is assumed that the generated revenue is absorbed externally. Including the sellers into the agent set yields a* combinatorial exchange*:*

Fix a set I of agents, a set J of goods and for every good $j \in J$, an agent seller(j). Define

$$X = \left\{ \left(k_j^i : i \in I, j \in J\right), k_j^i \in \{-1, 0\} \text{ for } i = \textit{seller}(j), k_j^i \in \{0, 1\} \text{ for } i \neq \textit{seller}(j), \right.$$
$$\left. \sum_{i \in I} k_j^i \leq 0 \text{ for all } j \in J \right\} \tag{2.39}$$

This models an exchange market with unique goods that are offered by one seller each (with possibly one seller selling different goods), and buyers that purchase a combination of goods. The restriction $\sum_{i \in I} k_j^i \leq 0)$ says that no more goods can be purchased than are sold, while it is allowed that goods are "left over".

Note that the Myerson-Satterthwaite theorem (Theorem 26) implies that there is no budget-balanced efficient mechanism with voluntary participation for X. In particular, the VGC mechanism is not budget balanced. In chapter 3 we will develop an adoption of VGC that is budget-balanced and satisfies voluntary participation (but is not efficient). In chapter 4, we will apply this mechanism to a setting were users submit competetive bids for a overlapping pathes through a network of links with limited capacity.

2.3 Example: A public project

Suppose there is a project that can be implemented on different levels. Given level x, let $c(x)$ be the cost incurred by the implementation of that level, where $c : X \mapsto \mathbb{R}^+$ is strictly monotonously increasing, and $c(0) = 0$. Now let there be agents i that benefit from the project, day, $u_i(x)$ is the utility of agent i from project level x. Suppose that $u_i(0) = 0$, and the u_i are monotonously increasing. Let $\mathcal{P}$ be the set of *user profiles*, that is, of all vectors $(u_i : i \in I)$.

Consider first the case that $X = \{0, 1\}$, that is, either the project is implemented, or not. A mechanism $\mathfrak{M}$ for that problem is a tuple $\mathfrak{M} = (x, p)$, where

- $x : \mathcal{P} \mapsto X$ is the decision function (the project is implemented if agents submit a profile $\vec{u}$ with $x(\vec{u}) = 1$), and
- p is a payment function with the property that if $x(\vec{u}) = 0$, then $p(\vec{u}) = \vec{0}$.

We claim that

Theorem 36. *Let $\mathfrak{M} = (x, p)$ be individual rational mechanism with truthtelling as dominant strategy for all players. Then there is a price vector $\vec{p} = (p_i : i \in I)$ with $p_i \in \mathbb{R}^+ \cup \{\infty\}$ such that for all user profiles $\vec{u} = (u_i)$,*

- *$x(\vec{u}) = 1$ if and only if for all i, $u_i \geq p_i$, and*
- $$p(\vec{u}) = \begin{cases} 0 & \textit{if } x(\vec{u}) = 0, \\ (p_i : i \in I) & \textit{otherwise.} \end{cases} \tag{2.40}$$

Proof. Let us write, for any i and $\vec{v} = (v_i : i \in I)$ and p_i,

$$P_i^+(\vec{v}, p_i) = \{(v_1, \ldots, v_{i-1}, q_i, v_{i-1}, \ldots, v_n) : q_i \geq p_i\} \tag{2.41}$$
$$P_i^-(\vec{v}, p_i) = \{(v_1, \ldots, v_{i-1}, q_i, v_{i-1}, \ldots, v_n) : q_i < p_i\}. \tag{2.42}$$

Let $\mathfrak{M} = (x, p)$ be truthful and individual rational. Let $X = \{\vec{v} : x^{\mathfrak{M}}(\vec{v}) = 1\}$.

Suppose that $x(\vec{v}) = 1$ and let $p(\vec{v}) = \vec{p} = (p_i)$. Then for any i,

$$P_i^+(\vec{v}, p_i) \subseteq X \tag{2.43}$$
$$P_i^-(\vec{v}, p_i) = \emptyset. \tag{2.44}$$

Let $\vec{u} \geq v$. Then $\vec{u} \in X$. We claim that $p(\vec{u}) = \vec{p}$. Suppose not.

Case 1: for some i, $(p(\vec{u}))_i > p_i$.

But then for $p_i < p_i^* < (p(\vec{u}))_i$, we have

$$(v_1, \ldots, v_{i-1}, p_i^*, v_{i+1}, \ldots n_n) \in P_i^+ \tag{2.45}$$

but this point is not in X, a contradiction.

Case 2: for all i, $(p(\vec{u}))_i \leq p_i$, and for some i strict $<$ holds.

Let $\vec{p}^* = p(\vec{u})$. Now on the one hand, $P_i^-(\vec{v}, p_i) \cap P_i^+(\vec{u}, p_i^*) \neq \emptyset$, but on the other hand, $P_i^+(\vec{v}, p_i^*) \subseteq X$, a contradiction. This finishes the proof. □

Let us also note that

- the mechanism that *always* implements the project and splits the costs by an arbitrary rule is budget balanced but not individually rational, and
- the mechanism that *always* implements the project and lets no one pay anything is individual rational but not budget balanced, and that
- for both these mechanisms, truth-telling is weakly dominant.

2.4 Example: Auctioning a divisible good

Suppose there is a resource of quantity 1 that can be split arbitrarily between different clients (agents). Let there be clients $i \in I$, then the set of possible allocations is $X = \{(x_i : i \in I) : x_i \geq 0, \sum_{i \in I} x_i \leq 1\}$. Assume that the utility functions $u_i : [0,1] \mapsto \mathbb{R}^+$ of the clients are monotonously increasing. Let $\mathfrak{M} = (o^{\mathfrak{M}}, p^{\mathfrak{M}})$ be a mechanism for X and $(u_i : i \in I)$ and write for a vector $\vec{u}$ of utility functions, $x_i^{\mathfrak{M}}(\vec{u}) = \left(o^{\mathfrak{M}}(\vec{u})\right)_i$ and $\vec{x}^{\mathfrak{M}}(\vec{u}) = (x_i^{\mathfrak{M}}(\vec{u}) : i \in I)$, dropping the superscript $\mathfrak{M}$ if no ambiguity arises. The efficiency condition (2.12) from definition 14 then takes the form

$$(\forall \vec{u})\, \vec{x}^{\mathfrak{M}}(\vec{u}) \in \arg \max_{(x_i : i \in I) \in X} \sum_{i \in I} u_i(x_i) \tag{2.46}$$

We write $V^{\mathfrak{M}}(\vec{u}) = \sum_{i \in I} u_i(x_i^{\mathfrak{M}}(\vec{u}))$. As in the previous section, if $\mathfrak{M}$ satisfies voluntary participation and the no positive transfers condition, the payment function has to be

$$p_i^{\mathfrak{M}}(\vec{u}) = V^{\mathfrak{M}}(\vec{u}|_{u_i \equiv 0}) - V^{\mathfrak{M}}(\vec{u}) + u_i(x_i^{\mathfrak{M}}(\vec{u})) \tag{2.47}$$

which we write as

$$p_i^{\mathfrak{M}}(\vec{u}) = u_i(x_i^{\mathfrak{M}}(\vec{u})) - \Delta_{\text{vic}}^{\mathfrak{M}}(\vec{u}) \tag{2.48}$$

3 A combinatorial exchange for autonomous traders

3.1 Introduction

Part of the work presented in this chapter was published in [76] and [74].

Internet auctions are one of the success stories of electronic commerce. Retrospectively, this is altogether not surprising, as modern economy produces a large turnover of goods; and a large quantity of high value goods is not allocated efficiently by traditional retail commerce but rather sold for giveaway prices. Apparently, there is a demand for a highly efficient secondary retail market, which specializes in transactions between partners that participate in the exchange market spontaneously. Auctions are an elegant way to tackle the problem of pricing and, properly used, can lead to efficient allocation of goods.

Bidding on single goods reflects utilities without interdependencies. If bundling goods increases utility (*complementary utility*), or goods can substitute each other, bidding on single goods involves a risk of incomplete or redundant purchases. Combinatorial auctions allow bidders to express more complex utility functions. Winner determination and payment allocation for one-sided combinatorial auctions is possible using the Vickrey-Grove-Clarke (VGC) mechanism. There is much work about complexity issues of VGC mechanism [21, 62, 66, 67]. While the exact problem is computationally intractable, there are approximation algorithms [87] with good stochastic performance and accuracy whose availability encourages us to leave complexity issues aside in this chapter. VGC leads to efficient, budget-balanced, individual rational, and even incentive compatible goods and payment distributions [62]. However, revealing true utilities looses much attractivity if bids under false names are possible. Sakurai et al. [64] and Yokoo et al. [83] show that there is no protocol with the properties stated above that is robust against false name bids. With these results in mind, Yokoo et al. [84] present a protocol which is budget-balanced, individual rational, and robust against false name bids, giving up on efficiency.

One-sided combinatorial auctions allocate goods from *one seller* to *many bidders*. For Internet auctions, we need to model a market with *many sellers* and *many bidders*. This is a special case of a *combinatorial exchange* or *double auction* [82]. The Vickrey-Groves-Clarke mechanism (VGC) when applied to combinatorial exchanges preserves all above properties except budget balance. Unfortunately, there is a grave negative result [56, 58] about protocols for combinatorial exchanges stating that there is no protocol for double auctions that is efficient, budget-balanced, and individual rational.

Internet auctions have a couple of peculiarities that are so far rarely considered in connection with double sided auctions:

- Both sellers and buyers enter their bids continuously.
- Sellers specify auction clearing times; there is no market inherent clearing rhythm.
- There is virtually no mean against false name bids.
- Sellers may wish to leave the pricing completely to the buyer's side, i.e., offer their goods without asking any specific price.
- Winner determination and payment allocation should benefit all individual traders.

We present protocols and algorithms for clearing, winner determination, and pricing double auctions in this setting which exhibit the following properties:

- They allow auctioneers to start auctions at any time and determine their life span.
- Bids can be aggregated, including combinatorial ones, over some time.
- A price for every successful auction is based solely on the collected bids.
- Bidding under false names is possible only with a risk of forfeiting trade.
- Pricing is budget-balanced and individual rational.

The rest of this chapter is structured as follows: Sect. 3.2 develops a combinatorial exchange model tailored for our application scenario. In section 3.3 we suggest some properties of payment allocation we consider necessary in the context of auctions with autonomous traders. In section 3.4, we present a new pricing algorithm that incorporates these properties. Section 3.5 presents a new bid clearing policy that we then show to perform superior in comparison with the "traditional" clearing policies.

Section 3.6 extends our pricing scheme to multiple item auctions – an extension that we will use in chapter 4 where we apply SBNL and commit window clearing to bandwidth reservation in networks.

3.2 A combinatorial exchange model for an auction platform application

We now introduce the notation used by our model which is a special type of allocation problem as in definition 32 and example 35. More precisely, our model describes a variant of a *clearing house* or *periodic double auction* that admits combinatorial bids. However, we impose a number of additional constraints that reduce the complexity of the model to make it more feasible in practice:

- Uniqueness of goods: exactly one copy of every good is being auctioned. This means that for every j, there is exactly one i (nameley, $i = \text{seller}(j)$) such that $k^i_j = -1$, and for all other i, $k^i_j = 0$ for $(k^i_j : i \in I, j \in J) \in X$.
- Only pure offers and pure buying bids. This implies that every agent i is either a *buyer* with $k^i_j \geq 0$ for all j, or a *seller* with $k^i_j \leq 0$ for all j. There are no agents that want to “exchange” one good for another.
- Only buying bids can be combinatorial, that is, for every seller i, there is exactly one good j with $k^i_j = -1$, and for all other all other j, we have $k^i_j = 0$.
- No substitutions – no OR-connected bids. This means that the buyer’s valuation functions are *convex*, that is, for every buyer i and sets of goods $J_1, J_2 \subseteq J$ with $J_1 \cap J_2 = \emptyset$, we have

$$u_i(J_1 \cup J_2) \geq u_i(J_1) + u_i(J_2) \tag{3.1}$$

- Free disposal is possible (i.e., the seller can keep his good), as modelled in equation 2.39.
- Price is computed only from buying bids, i.e., sellers do not specify any reservation price. This means that the seller’s valuation function is identical to 0.

We do not consider OR-connected bids because it is hard to mediate between interest conflicts arising between auctioneers when there are not enough bids to sell all items.

Our market trades with n goods (so $|J| = n$). Convexity of the valuation function and the assumption that free disposal is possible allows us to assume without loss of generality that every buyer *bids for exactly one bundle of goods*, that is, that there is for every buyer i a set $J(i) \subseteq J$ such that $u_i(J) > 0$ and for all $J' \neq J$, $u_i(J') = 0$. (So i would not appreciate “additional” goods even “for free”: we can use free disposal to avoid such “gifts”.)

We identify now agent i with his “bid” b and write $b = (k^b_1, \dots, k^b_n, p^b)$ with $k^b_i \in \{-1, 0, 1\}$ and $p^b \in \mathbb{R}$. We distinguish between *auction bids* (or *auctions*) and *buying bids*, respectively:

Auction bids represent the offered goods. For auction bids all k_i^b's are 0 except one, and this one has value -1. An auction bid b is *offering good* i if $k_i^b = 1$. Furthermore, we impose $p^b = 0$ for auction bids as any pricing is left to the buyers. This means that b is auctioning the good "with no price limits". As goods are unique, we assume that for any good i, there is exactly one auction bid a_i offering i.

Buying bids represent the demanded goods. For buying bids all k_i^b's have either value 0 or 1. A buying bid b is *bidding for goods* $\{i_1, \ldots, i_x\}$ if $k_i^b = 1$ for all $i \in \{i_1, \ldots, i_x\}$ and $k_i^b = 0$ otherwise. Here, p^b is always positive (due to the free disposal requirement negative bids are not reasonable) and represents the amount the buyer is willing to pay for the goods he is bidding for. For example, the buying bid $(0, 1, 1, 20)$ means that a buyer is willing to pay a maximum of 20 for goods 2 and 3.

Let $\mathcal{A}$ and $\mathcal{B}$ be the sets of auction and buying bids in the market, respectively.

Definition 37 (Winner determination algorithm). *A* winner determination algorithm *takes as input a set $\mathcal{A}$ of auction bids and a set $\mathcal{B}$ of buying bids. From this, it computes an* acceptance function $\chi : \mathcal{B} \mapsto \{0, 1\}$ *with*

$$\sum_{b=(k_1^b,\ldots,k_n^b,p^b)\in\mathcal{B}} \chi(b) \cdot k_i^b \leq 1 \tag{3.2}$$

for all $i = 1, \ldots, n$. A bid $b \in \mathcal{B}$ is accepted *by the algorithm if $\chi(b) = 1$, and* rejected *otherwise.*

Informally spoken, a winner determination algorithm determines for each offered good at most one buying bid that is accepted.

Definition 38 (Payment allocation algorithm). *A* payment allocation algorithm *takes as input a set $\mathcal{A}$ of auction bids, a set $\mathcal{B}$ of buying bids, and an acceptance function χ. From this, the algorithm computes a* payment allocation function $p : \mathcal{A} \cup \mathcal{B} \mapsto \mathbb{R}$.

A payment allocation function p satisfies budget-balance *if*

$$\sum_{c\in\mathcal{A}\cup\mathcal{B}} p(c) \geq 0$$

It is individual rational, *iff for all bids b with $\chi(b) = 1$ we have $p(b) \leq p^b$, and for all b with $\chi(b) = 0$ we have $p(b) = 0$.*

Hence, a payment allocation algorithm assigns to each accepted bid its corresponding revenue which is positive for a buying bid and negative for an auction bid. The following example describes the two-sided VGC mechanism from example 35 in our simplified notation:

Example 39 (Two-sided Vickrey-Groves-Clarke). *Let χ be maximizing the sum of revenues*

$$V^* = \sum_{b\in\mathcal{B}} \chi(b) \cdot p^b \tag{3.3}$$

subject to condition (3.2). Let for $c \in \mathcal{A} \cup \mathcal{B}$ be $(V_{-c})^*$ be the maximized sum of revenues for auctions and bids $\mathcal{A} \cup \mathcal{B} \setminus \{c\}$, and let

$$\Delta_{vick,c} = V^* - (V_{-c})^* \tag{3.4}$$

be the Vickrey discount *for c. Let the payment function p be defined by*

$$p(a) = -\Delta_{vick,a} \qquad \text{for } a \in \mathcal{A} \tag{3.5}$$

$$p(b) = p^b - \Delta_{vick,b} \qquad \text{for } b \in \mathcal{B}. \tag{3.6}$$

The resulting mechanism (χ, p) is the Two-sided Vickrey-Groves-Clarke mechanism for combinatorial exchanges.

The last constraint implies that sellers cannot specify a minimum price as precondition for selling their good. Note that while two-sided VGC does allow sellers to specify a negative utility for the sale of a good, this is not really a *minimum price condition* because specification of negative utility from a sale changes payment allocation even when more than the lost utility is given to the seller anyway, as demonstrated by the following example:

Example 40. *Let there be two auctions and one bid of 10 for both items together. Suppose first that the auctions are without minimum price.*

According to VGC, all auctions and bids would be matched, the following payments would be allocated: the auctioneers would receive a payment of 10 each, while the bidder's payment would be 0.

Suppose now that the first auctioneer would demand a minimum price of 7. VGC would then allocate a payment of 10 to this auctioneer, while the other one's payment would shrink to 7.

This contrasts with one-sided Vickrey payment or plain pay-your-bid payment for single item auctions where specifying a minimum price does not change payment allocation once the payment surpasses the minimum price.

In many settings, specification of minimum prices is not required by the sellers. In particular, this holds when the market has sufficient liquidity,

or when the fixed - variable cost ratio is high. We therefore refrain from considering minimum prices. We nevertheless acknowledge the problem of respecting minimum prices in the above sense, while preserving other desired properties of the payment allocation algorithm, an interesting question for further research.

3.3 Pricing properties required by autonomous traders

Pricing in a combinatorial exchange is far from being trivial. Following [58] and [39], we take individual rationality and budget-balance as hard constraints that our payment allocation algorithm must satisfy. Besides these two constraints, we consider a couple of other properties being useful which are discussed in the following.

3.3.1 Respecting single item bids.

By accepting combinatorial bids, we expect more willingness from the bidders to bid and therefore an increased total revenue. Thus, a reasonable constraint for the payment allocation is that no bidder looses from combinatorial bids:

Definition 41. *A payment allocation function* p respects single item bids*, if for all auctions* a *and for all buying bids* b *that bid only for* a*, we have* $p(a) \geq p^b$*.*

Proposition 42. *VGC respects single item bids.*

Proof. Let a be an auction bid, and b_a be a bid bidding for a only. Let V_{-a} be the maximized revenue of all auctions except a. By accepting bid b_a, the revenue increased by p^{b_a}. So a increases the total revenue by at least p^{b_a}, and therefore a's Vickrey discount is at least p^{b_a}. □

Parkes et al. [58] present some VGC-based budget-balanced payment rules. The rules are generated by minimizing the deviation from the Vickrey payments measured in various distance function. Practically, they divide the available revenue[1] between all traders, using various division rules:

- The *Equal* payment rule splits the available surplus equally among all sellers and buyers.
- The *Fractional* payment rule splits the available surplus according to the fractional share from the total Vickrey discount of every agent
- *Small* starts awarding discounts to the traders with small Δ_{vick} and proceeds until the available discount is used up.

While VGC does respect single item bids, these variants of VGC do not as is illustrated in the following example:

Example 43. *Let there be auctions and bids*

$$\begin{aligned} a_1 &: (-1, 0, 0) \\ a_2 &: (0, -1, 0) \\ b_1 &: (1, 1, 60) \\ b_2 &: (1, 0, 50) \\ b_3 &: (1, 0, 49) \end{aligned}$$

a_1, a_2 *and* b_1 *are accepted. The available surplus is 60. The Vickrey discounts for the agents are:*

$$\begin{aligned} a_1 &: 60 \\ a_2 &: 10 \\ b_1 &: 10 \end{aligned}$$

[1]Remember that we have no minimum prices in our setting

The Equal *payment rule splits the available surplus equally, so a_1 and a_2 receive 20 each, and b_1 pay 40. However, a_1 would prefer to accept bid b_2 with a surplus of 50, leaving a_1 with a share of 25 under the* Equal *payment rule. The* Fractional *payment rule leads to the following payments: a_1 receives $60 \cdot 60/80 = 45$, a_2 receives $10 \cdot 60/80 = 7.5$, b_1 pays $60 - 10 \cdot 60/80 = 52.5$. If however a_1 accepts bid b_2, a surplus of 50 results. The Vickrey discount of a_1 is 50, of b_2 is only 1, and a_1 receives a payment of $50 \cdot 49/50 = 49$ under the* Fractional *rule. Similarly, examples for the other payment rules (*Threshold, Small, Large, *and* Reverse *payment) can be constructed showing that they do not respect single item bids.*

We are tempted to generalize single item bid respect to "all bids respect" by demanding that for all bids b

$$\sum_{1 \leq i \leq n} k_i^b \cdot p(a_i) \geq p^b \tag{3.7}$$

where a_i is the auctioning bid of the auction offering good i. Basically, this would mean that every auction can choose its favourite bid to be accepted. However, we can easily see that this is incompatible with budget-balance:

Example 44. *Let there be three auctions, and let there be bids as follows:*

$$\begin{aligned} b_1 &: (1,1,0,10) \\ b_2 &: (1,0,1,10) \\ b_3 &: (0,1,1,10) \end{aligned}$$

The maximal revenue is 10 as only one bid can be accepted. To satisfy the three inequalities resulting from (3.7), we would need a revenue of 15, however.

3.3.2 No loss from a bid.

Next, we desire that no auctioneer ever looses from a bid for his good. Formally, that means:

Definition 45. *A payment allocation algorithm has the* no loss from a bid *property if the following holds: Let $\mathcal{A}$ be a set of auctions and let $\mathcal{B}$ be a set of bids. Let $a \in \mathcal{A}$ be an auction offering good i and let $b = (k_1^b, \ldots, k_n^b, p^b)$ be a bid with $k_i^b = 1$. Let p be the payment allocation function for $(\mathcal{A}, \mathcal{B})$ and let p' be the payment allocation function of $(\mathcal{A}, \mathcal{B} \cup \{b\})$. Then $p'(a) \geq p(a)$.*

Note that VGC does satisfy the no loss from a bid property. However, the *Small* rule of [58] does not:

Example 46. *Let there be two auctions a_1 and a_2, and bids as follows:*

$$\begin{aligned} b_1 &: (1, 1, 100) \\ b_2 &: (1, 1, 99) \\ b_3 &: (1, 0, 1) \end{aligned}$$

Then bid b_1 is accepted, $V^* = 100$, and

$$\begin{aligned} \Delta_{vick,a_1} &= 100 \\ \Delta_{vick,a_2} &= 99 \\ \Delta_{vick,b_1} &= 1 \end{aligned}$$

and the Small *rule allocated discounts to b_1 and a_2, leaving a_2 with a payment of 99 and a_1 with no payment. Suppose now that there is an additional bid*

$$b_4 : (0, 1, 2)$$

Now the discount goes to b_1 and a_1, leaving a_2 with no payment. So a_2 suffered from an additional bid.

3.4 A new pricing scheme

Now, we present a budget-balanced, individual rational, single item bid respecting payment allocation algorithm with the no loss from a bid property.

Algorithm SBNL

Input: $\mathcal{A}$ – a set of auctions, $\mathcal{B}$ – a set of bids
Output: a payment allocation function $p : \mathcal{A} \mapsto \mathbb{R}$.

- *Step 1.* Compute the item allocation that maximizes revenue. Let V be the maximized revenue.
- *Step 2.* For an auction a offering good i, let $b_a = (0, \ldots, 0, 1, \ldots, 0, p^{b_a})$ be the highest bid bidding for good i only. If there is no such a bid, define $p^{b_a} = 0$. Let $V_{\text{single}} = \sum_{a \in \mathcal{A}} p^{b_a}$.
- *Step 3.* Solve the linear programming problem

$$\begin{aligned} &\text{Minimize} && Y = \sum_{1 \le i \le n} y^i \\ &\text{such that} && \\ &(\forall_{b \in \mathcal{B}}) && \sum_{1 \le i \le n} k_i^b \cdot y^i \ge p^b \end{aligned}$$

Among all optimal $(y^i : 1 \le i \le n)$, choose the one that minimizes $\sum_i (y^i)^2$.

- *Step 4.* Let $Q = \frac{V - V_{\text{single}}}{Y - V_{\text{single}}}$.
- *Step 5.* For all auctions $a \in \mathcal{A}$, let $p(a) = p^{b_a} + (y^i - p^{b_a}) \cdot Q$, where a is offering good i.

Our pricing mechanism let successful buyers pay exactly the amount of their bid. Winner determination takes place subject to maximizing total revenue. The pricing mechanism distributes this revenue among the sellers.

Proposition 47. *Algorithm SBNL satisfies budget-balance and individual rationality, respects single item bids, and has the no loss from a bid property.*

Proof. Obviously the algorithm is individual rational. The sum of the payments is

$$\sum_a p(a) = \sum_a p^{b_a} + Q \cdot \left(\sum_i y^i - \sum_a p^{b_a}\right) = V_{\text{single}} + Q \cdot (Y - V_{\text{single}}) = V$$

and this proves budget-balance. Step 2 ensures single item bid respect. For the no loss from a bid property, note that we always have $Y \geq V$, and this implies $Q < 1$. Thus an additional single item bid for a can only increase a's payment. The argument for an additional combinatorial bid is similar.

□

3.5 Bid synchronization

After developing a pricing scheme, we will now turn our attention to the *clearing policy* of an auction protocol which defines at what times auction and buying bids are being matched.

Our market model allows continuous publication of new auctions. There are various clearing strategies in use for continuous double auction markets [19, 20]:

- *Continuous clearing.* The trade occurs as soon matching bids and asks arrive.
- *Periodic or random clearing.* The trade occurs at certain times (periodic, random, or a combination of both), bids and asks are matched subject to certain optimality conditions (e.g. maximizing surplus or throughput).

All three policies have serious drawbacks in our scenario. Periodic clearing of bids that are submitted continuously results in auctions whose live span is very short when they are entered shortly before the end of an aggregation slot. A similar effect occurs with random clearing: some auctions will close after a short time, and little value is generated, while others may run longer than the auctioneer desires to wait. Continuous clearing can't be used when sell-bids have no minimum price, since otherwise, sell-bids would always be matched with the first bid that asks for the offered good.

We find it desirable to allow the auctioneer to control the live span of his auction. Therefore, we use another clearing policy that we now describe.

3.5.1 A new clearing policy

Definition 48 (Commit window clearing)**.** *Every auction announcement*

$$\mathbf{a} = (a^{\mathbf{a}}, t^{\mathbf{a}}_{earliest}, t^{\mathbf{a}}_{latest}) \tag{3.8}$$

contains the following information:

- *the auction bid $a^{\mathbf{a}}$, i.e. identity of the good*
- *the earliest commit time $t^{\mathbf{a}}_{earliest}$*
- *the latest commit time $t^{\mathbf{a}}_{latest}$*

Every auction goes through the following sequential phases:

- Pre-commit. *Bids for this auction can be submitted. The auction will not commit to accepting any of them.*
- Allow-commit. *Bids for this auction can be submitted. The auction house can request that the auction commits to a bid if that bid wins by the winner determination algorithm applied by the auction house. In this case, all unsuccessful bids for this auction are uncommitted, and the auction transits into Expired state.*
- Force-commit. *No bids can be submitted anymore for this auction. The winner determination algorithm determines the winner among all bids that bid for auctions in Allow-commit or Force-commit stage. Non-accepted bids for this auction are uncommitted. Transit into Expired state.*
- Expired. *The auction is finished, the winner was determined and the payment computed.*

A bid is committable *if all auctions the bid is bidding for are in Allow-commit or Force-commit state. Pre-commit for an auction $\mathbf{a}$ is the time before $t^{\mathbf{a}}_{earliest}$. The Allow-commit phase lasts from $t^{\mathbf{a}}_{earliest}$ to $t^{\mathbf{a}}_{latest}$ and are followed by the Force-commit and Expired phases.*

This policy lets the auctioneer control the live span of his auction. A combinatorial bid can be accepted if the commit phases of all auctions bidden for do overlap. The larger the Allow-commit phase, the more inviting his auction will be toward combinatorial bids.

3.5.2 A lower bound for the revenue

In this section, we give a lower bound of the revenue assuming that the set of auctions open for bidding does not change. In this case, it is reasonable to assume that the set of items bidden for and the amount of the bids submitted within that interval are independently, but identically distributed.

Let us fix a set $\mathcal{A}$ of open auctions $\mathcal{A} = \{A_1, ..., A_n\}$ and assume that bids are independent random variables $b(p, \mathcal{A}, \#^2)$ for fixed $p \in [0, 1]$.

Consequently, the number of items a bid b is bidding for follows binomial distribution with parameters n and p:

$$\#\text{items}(b) \sim \text{Binomial}(n, p)$$

and the amount distribution is a squared binomial distribution.

Let us compute the expected value of the revenue generated by one bid.

$$\begin{aligned} E[\text{amount}(b)] &= \sum_{k=0}^{n} \binom{n}{k} p^k (1-p)^{n-k} k^2 \\ &= n(1-p)^n p(1-p+np)(1+\frac{p}{1-p})^n \end{aligned} \tag{3.9}$$

The cumulative distribution function is

$$P(\text{amount}(b) \leq x) = \sum_{k=0}^{\sqrt{x}} \binom{n}{k} p^k (1-p)^{n-k},$$

and thus

$$\begin{aligned} &P(\max\{\text{amount}(b_1), ..., \text{amount}(b_g)\} \leq x) \\ &\quad = (P(\text{amount}(b) \leq x))^g \\ &\quad = \left(\sum_{k=0}^{\sqrt{x}} \binom{n}{k} p^k (1-p)^{n-k} \right)^g \end{aligned} \tag{3.10}$$

Now we can give a ***lower bound*** $E_{\max}(g, n, p)$ on the expected value $E_g(n, p)$ of the revenue of the auctions with g bids

$$\begin{aligned} E_g(n, p) &= E(\text{revenue of } g \text{ bids}) \\ &\geq E(\max\{\text{amount}(b_1), ..., \text{amount}(b_g)\}) \\ &= \sum_{x=0}^{n^2} \left(1 - \left\{ \sum \binom{n}{k} p^k (1-p)^{n-k} \right\}^g \right) \\ &= E_{\max}(g, n, p) \end{aligned} \tag{3.11}$$

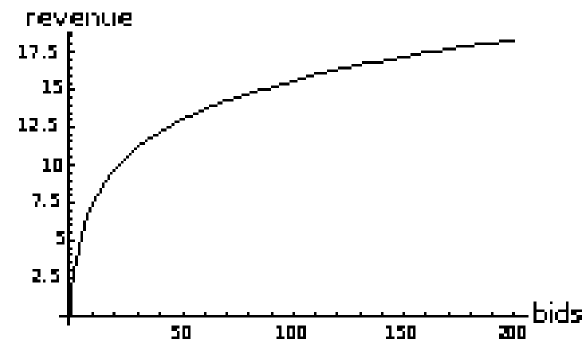


Figure 3.1: Maximal revenue of one bid over number of bids.

Figure 3.1 is a plot of values for $E_{\max}(g, n, p)$ with n=10, p=0.1, $0 \leq g \leq 200$.

Now if *0<p<1*, $P(\text{amount}(b) \leq x) = 1$ for $x \geq n^2$ and $P(\text{amount}(b) \leq x) < 1$ for $x < n^2$ and therefore,

$$\lim_{g \to \infty} E(\max\{\text{amount}(b_1), ..., \text{amount}(b_g)\}) = n^2$$

On the other hand, the revenue of the nth auctions is bounded from above by n^2. Thus we conclude that $\lim_{g \to \infty} E_g(n, p) = n^2$ for all *0<p<1*.

The lower bound we gave is not tight at all. We are not aware of a closed-form representation of the precise expected revenue $E_g(n, p)$. Figure 3.2 shows results of a numeric simulation of $E_g(10, 0.1)$ with 1000 iterations per $g(0 \leq g \leq 20)$, plotted over the lower bound $E_{\max}(g, n, p)$.

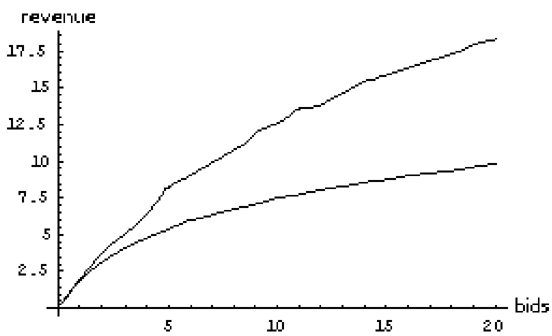


Figure 3.2: Simulated revenue and lower bound over number of bids.

3.5.3 Comparison between periodic and commit window clearing

3.5.3.1 Side conditions for the comparison

We wish to evaluate efficiency in connection with the surplus-maximizing good allocation of two of the three mentioned clearing policies: the new commit window clearing policy, and periodic clearing. We will measure efficiency in terms of the generated revenue[2] *per auction*, for a fixed set of auctions the timing of which we adjust to the used clearing policy.

For periodic and random clearing, the time when bids and offers are matched is determined by the clearing policy parameters - neither auctions nor bids have to state anything about that time. Let Δ_{periodic} be the length of the interval between two clearings for the periodic clearing policy. Then the length of an auction is between 0 and Δ_{periodic}.

For commit window clearing, the length of an auction is between t_{earliest} and t_{latest}. Its precise value will be determined during the life of the auction and will depend on the submitted bids.

We will model behaviour of market participants as follows:

- Sellers initiate new auctions according to a Poisson process with parameter λ_a. Sellers require that these auctions must terminate within a specified time $t_{\text{MaxAuctionDuration}}$ which is constant for all auctions. For a given auction, let t_{start} be its start time.

- Sellers have no fixed costs and therefore, wish to auction their goods without minimum price.

- For the periodic auction, we set $\Delta_{\text{periodic}} = t_{\text{MaxAuctionDuration}}$. The auction lives from t_{start} to the end of the current clearing period, that is to $\min\{n\Delta_{\text{periodic}} : n \in \mathbb{N}, n\Delta_{\text{periodic}} \geq t_{\text{start}}\}$.

- For the commit window clearing, we set $t_{\text{latest}} = t_{\text{start}} + t_{\text{MaxAuctionDuration}}$. The size of the commit window s_{window} can vary from 0 to $t_{\text{MaxAuctionDuration}}$ and therefore, t_{earliest} varies from t_{start} to $t_{\text{start}} + t_{\text{MaxAuctionDuration}}$. For our study, we fix the commit window size to its maximal value $s_{\text{window}} = t_{\text{MaxAuctionDuration}}$.

- Buyers submit bids for combinations of goods. A Poisson process with parameter λ_b determines the times when a bid is submitted. A bid, submitted at time t_{bid}, will be a random variable $b(p, \mathcal{A}, \#^2 + \delta)$ where $\mathcal{A}$ is the set of auctions open for bidding at the time when the bid is submitted, and

$$\delta \sim N(0, \#^{-1}) \tag{3.12}$$

[2]social surplus

> In the choice of the amount distribution, we follow [9] who suggests as an acceptable distribution of the bid amount a normal distribution with expected value equaling the "fair" value of the item combination bidding for.

We claim that commit window clearing generates, under the side conditions described above and for s_{window} properly chosen, a higher revenue than periodic clearing. We support this claim by some simulations whose parameters are derived from mentioned side conditions.

3.5.3.2 Simulation results

Figure 3.3 shows a comparison of revenues of auctions with periodic and commit window clearing. The parameter p varies in 100 steps between 0 and 1. The Poisson parameters λ_a and λ_b are constants with value 0.1. The auction duration is set to 100. The average total revenue of 50 iterations for all auctions generated within a period of 10000 was measured.

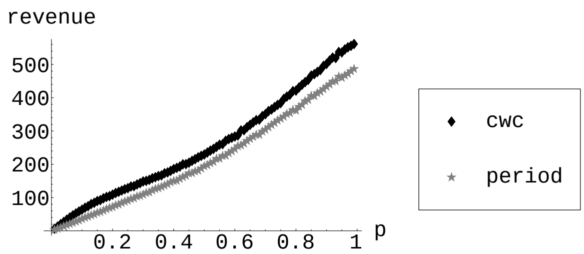


Figure 3.3: Comparison of periodic and commit window clearing

We conclude that this simulation supports our claim.

3.5.3.3 An analytic approach for offline winner determination

Bids in the stochastic model from above are generated on the fly with a target chosen from all auctions that are open for bidding at the time when a bid is submitted. In this section, we will modify bid generation slightly: the target of a bid submitted at time t_{bid} is chosen from all auctions a with $t_{\text{earliest}} \leq t_{\text{bid}} \leq t_{\text{latest}}$. Winner determination takes place offline, that is, among *all* bids that were submitted during the total run. For this scenario, we can show analytically that commit window clearing generates a higher surplus, then periodic clearing.

For a set $\mathcal{B}$ of bids bidding for subsets of the auction set $\mathcal{A}$, define

$$c(\mathcal{B}) = \begin{cases} 1 \text{ if all bids is } \chi \text{ are compatible} \\ \text{0 otherwise.} \end{cases} \tag{3.13}$$

and let revenue($\mathcal{B}$) be the maximized revenue possible to generate from $\mathcal{B}$, that is,

$$\text{revenue}(\mathcal{B}) = \max_{\mathcal{B}' \subseteq \mathcal{B}} c(\mathcal{B}') \sum_{b \in \mathcal{B}'} \text{amount}(b). \tag{3.14}$$

The following is easy:

Lemma 49. *Let there be given two finite sets of bids, $\mathcal{B} = \{B_i : i \in I\}$, and $\mathcal{B}' = \{B'_i : i \in I\}$, with the property that for every $i, j \in I$,*

- *amount(s_i) = amount(s'_i), and*
- *if s_i and s_j are compatible, so are s'_i and s'_j.*

Then

$$\textit{revenue}(\mathcal{B}) \leq \textit{revenue}(\mathcal{B}') \tag{3.15}$$

□

This implies

Lemma 50. *Let $\mathcal{S} = \{b_i : i \in I\}$ be a set of independently identically distributed variables*

$$b_i \sim b(p, \mathcal{A}, \#^2) \tag{3.16}$$

and let $\mathcal{S}' = \{b'_i : i \in I\}$ be a set of independently identically distributed variables

$$b'_i \sim b(p, \mathcal{A}_i, \#^2) \tag{3.17}$$

for some $\mathcal{A}_i$ with $|\mathcal{A}_i| = |\mathcal{A}|$.

Then

$$E(\textit{revenue}(\mathcal{S})) \leq E(\textit{revenue}(\mathcal{S}')) \tag{3.18}$$

□

Suppose auctions are started with constant rate r_a during time T and bids submitted with constant rate r_b (number per second). For simplicity, let r_b be a multiple of r_a. Furthermore, assume $t_{\text{MaxAuctionDuration}}$ is constant, and the size of the commit window $s_{\text{window}} = t_{\text{MaxAuctionDuration}} = \Delta_{\text{periodic}}$.

Figure 3.4 illustrates the situation with commit window clearing: auctions A_1 to A_{11} are subsequently started. Auctions A_i and A_j overlap if $|i-j| \leq \frac{\text{MaxAuctionDuration}}{r_a}$. For three bids B_1, B_2 and B_3, vertical lines show the auctions the bid target is drawn from.

The situation for the periodic clearing is shown in figure 3.5. The target of all bids is contained in the set of auctions started within one clearing interval.

Now let the b_i^{per} be, for $1 \leq i \leq r_b T$, independent, identically distributed random variables with distribution $b(p, \{A_{[\frac{i}{r_b \Delta_{\text{periodic}}}]+1}, \ldots, A_{[\frac{i}{r_b \Delta_{\text{periodic}}}]+r_a \Delta_{\text{periodic}}}\}, \#^2)$, and let

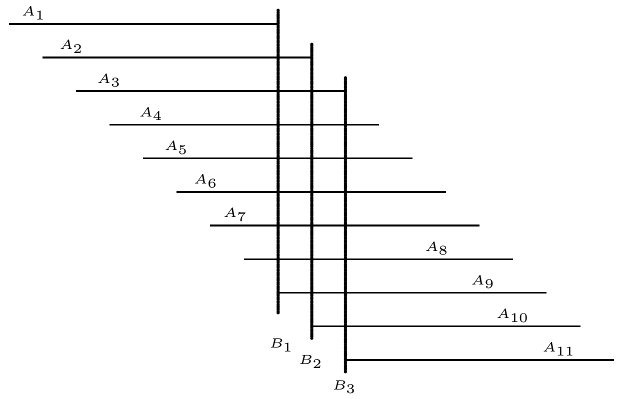


Figure 3.4: Overlapping auctions for commit window clearing

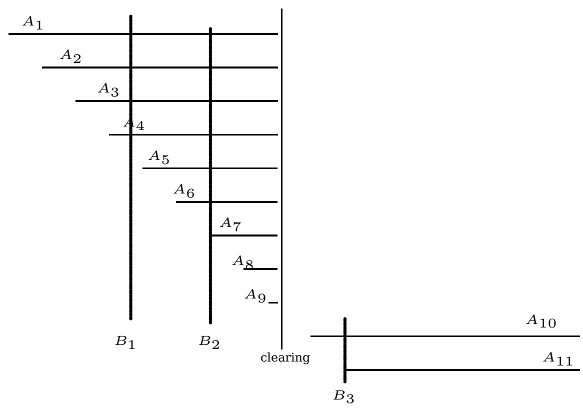


Figure 3.5: Auctions for periodic clearing

b_i^{cwc} be, accordingly, independent, identically distributed random variables with distribution $b(p, \{A_{i\frac{r_a}{r_b}}, \ldots, A_{i\frac{r_a}{r_b}+\text{MaxAuctionDuration}\cdot r_a}, \#^2)$. We compute

$$
\begin{aligned}
&E(\text{revenue of periodic clearing}) \\
&\quad = \text{revenue}(\{b_i^{\text{per}} : 1 \leq i \leq r_b T\})) && (3.19) \\
&\quad \leq \text{revenue}(\{b_i^{\text{cwc}} : 1 \leq i \leq r_b T\})) && (3.20)
\end{aligned}
$$

where the last inequality follows from lemma 50.

We conclude

Corollary 51. *The expected revenue of a commit window auction with maximal auction duration $t_{\text{MaxAuctionDuration}}$, auction start rate r_a and bid submission rate r_b is greater or equal to the expected revenue of a periodic auction with $r_a t_{\text{MaxAuctionDuration}}$ auctions running in parallel and $r_b t_{\text{MaxAuctionDuration}}$ bids.*

3.5.3.4 Further research on the performance of commit window clearing

Here we presented a first analysis on the efficiency of a new clearing policy suitable for combinatorial exchanges with multiple sellers and buyers. The policy was compared with the classical periodic clearing policy, and it was found that commit window clearing yields a higher mean revenue when auction and bid submission rates, bid distribution and maximal auction duration are fixed.

Some estimates on the expected revenue for periodic and commit window clearing were presented.

Undoubtedly, further analysis of the policies in regard of their generated revenue, particularly under the side conditions used for the simulation, would be quite interesting. The same should apply to analyzing more clearing policies like random clearing, and generalizing the results to more amount distribution functions.

3.6 Extending SBNL to multiple item auctions

Remember that bids are of the form $b = (k_1^b, \ldots, k_n^b, p^b)$, where we defined above that $k_i^b \in \{-1, 0, 1\}$. For buying bids, we have $k_i^b \in \{0, 1\}$, while for auction bids, $k_i^b \leq 0$. We now generalize the notion of auction bids by allowing $k_i^b \in \mathbb{Z}^-$, the set of non-positive integers. We keep the assumption that auction bids are non-combinatorial, that is, that for every bid b, there is only one i with $k_i^b \neq 0$. Now we can generalize definition 37.

Definition 52 (Multiple item winner determination algorithm)**.** *A* multiple item winner determination algorithm *takes as input a set $\mathcal{A}$ of auction bids*

$$
\mathcal{A} = \left\{ b_a = (k_1^{b_a}, \ldots, k_n^{b_a}, p^{b_a}) : b_a \in \mathcal{A} \right\} \tag{3.21}
$$

such that for every i, there is exactly one b_a with $k_i^{b_a} \neq 0$, and a set $\mathcal{B}$ of buying bids. From this, it computes an acceptance function $\chi : \mathcal{B} \mapsto \{0,1\}$ *with*

$$\sum_{b=(k_1^b,\ldots,k_n^b,p^b)\in\mathcal{B}} \chi(b) \cdot k_i^b \leq -k_i^{b_a} \tag{3.22}$$

for all $i = 1,\ldots,n$, where b_a is the unique bid in $\mathcal{A}$ with $k_i^{b_a} \neq 0$. A bid $b \in \mathcal{B}$ is accepted *by the algorithm if $\chi(b) = 1$, and* rejected *otherwise.*

Definition 38 remains unchanged. It is easy to see that the two-sided Vickrey-Groves-Clarke mechanism of definition 39 can be generalized to multiple item auctions:

Definition 53 (Two-sided Vickrey-Groves-Clarke for multiple item auctions)**.** *Let χ be maximizing the sum of revenues*

$$V^* = \sum_{b\in\mathcal{B}} \chi(b) \cdot p^b \tag{3.23}$$

subject to condition (3.22). Let for $c \in \mathcal{A} \cup \mathcal{B}$ be $(V_{-c})^*$ be the maximized sum of revenues for auctions and bids $\mathcal{A} \cup \mathcal{B} \setminus \{c\}$, and let

$$\Delta_{\mathit{vick},c} = V^* - (V_{-c})^* \tag{3.24}$$

be the Vickrey discount *for c. Let the payment function p be defined by*

$$p(a) = -\Delta_{\mathit{vick},a} \qquad \text{for } a \in \mathcal{A} \tag{3.25}$$

$$p(b) = p^b - \Delta_{\mathit{vick},b} \qquad \text{for } b \in \mathcal{B}. \tag{3.26}$$

The resulting mechanism (χ, p) is the Two-sided Vickrey-Groves-Clarke mechanism *for combinatorial exchanges with multiple items.*

It follows that the mechanism SBNL generalizes to the multiple item case as well.

4 Application to network management: Advanced resource reservation in networks

Nisan and Ronen define in [57, p.13] a scenario of a network that consists of a directed graph G whose nodes are connected by edges that represent links with associated costs for usage. In this setting, they define the natural shortest path problem that now corresponds to fining the cost-minimizing path between two nodes. A Vickrey mechanism in which the link costs are treated as bids, gives the "owners" of the links incentive to bid according to their true costs. Nisan and Ronen's results are extended by Hershberger and Suri [32] and Archer and Tardos [5]. Hershberger and Suri proved that the Vickrey pricing can be computed in the same time as the solution of a single-source shortest path problem, if the graph is undirected[1]. For results on directed graphs, see the follow-up paper [35].

The Vickrey mechanism pays more than the actual costs to the link owners (thus buying truthfulness). Archer and Tardos note that the additional *premium* can be a multiple of the actual costs. They prove that for a large class of graphs, every truthful mechanism has in some instances to pay a high premium even if there is a choice between multiple paths of essentially the same cost.

We have argued above that the fact that marginal usage costs are neglectible compared to fixed costs makes it disputable whether there is a meaningful definition of "costs" associated with usage of a link. Similar to the application scenario for transport logistics, it seems much more natural to leave the pricing to the *network users* that compete for resources. Users, however, would bid for *paths* rather than for single links. The resulting market would be precisely the combinatorial exchange that this chapter deals with.

Are the *pricing rules* of SBNL suitable for such a scenario?

- The requirement of *budget balance* may be obsolete if payments are small compared with fixed "basic fees" users pay to the network owners independent from usage. A pricing structure that is split into fixed fees and per-usage fees is appropriate if the basic fee users pay is mirrored by a *basic utility* users

[1] Beware of the conference version [33] of that paper, which erroneously claims that that holds for directed graphs, too (see also the erratum [34] of the authors).

have from being “connected” to the network. The amount of the fixed fee would then be determined independently from the exchange pricing. The bids of the users would be based on the *marginal utility* users have from using a certain path. Of course, there would be then a question on the acceptance of the fixed fee by the user. Thus, there would be no clear separation of the analysis of the mechanism, and the analysis of the economical context.

In the absence of basic fees, however, budget balance of the exchange market would be strictly required. This would hold true even if there is a basic fee but per-usage fees may be of comparable size.

- *Respecting single item bids* and *no loss from a bid* are clearly desirable properties from the point of view of the link owners.

We conclude that if budget balance is required, SBNL’s pricing rules are reasonable choice for network linkage pricing.

The definition of good synchronization rules seems more challenging. We have so far not yet stated whether the bid amount refer to a payment per package or per time interval. Nisan and Ronen have left that question unconsidered.

- One approach is to introduce *time slots* and auction them each to one bidder exclusively. This makes sense if the link’s capacity can’t be split between bidders. If this is not the case, we could divide the link’s capacity into smaller portions and auction them either as independent goods, or as a good with quantity more than one. But from the bidder’s point of view, the portions are substitutes, so if they are auctioned independently, we would have to allow bidders to submit OR-connected bids. In the second case, the auction would be a multi-item auction.

- Alternatively, we can auction the right to *send a package over the link within a time slot*. The link’s capacity will most probably allow more than one package per time slot. This means that the *uniqueness of goods* condition defined above is violated.

Table 4.1 gives a summary of the possible market types.
How do the second and third lines of the table relate to each other?

Proposition 54. *AND-of-OR-connected bids have strictly greater expressive power than selling bids for identical copies: That is, let there be a combinatorial exchange market AND-OR with*

- selling bids *of the form* $b_s = (\boldsymbol{slot}, \boldsymbol{link}, \boldsymbol{portion})$ *where*
  - ***slot*** *is a time slot,*

Unsplittable link capacity	Combinatorial exchange with AND-connected buying bids and non-combinatorial selling bids
Link capacity portions as distinct goods	Combinatorial exchange with AND-of-OR-connected buying bids and non-combinatorial selling bids
Link capacity portions as identical copies of the same good	Combinatorial exchange with AND-connected buying bids and selling bids with quantity
Package in a timeslot auction	Combinatorial exchange with AND-connected buying bids and selling bids with quantity

Table 4.1: Possible market types for network bandwidth auctions

- ***link*** *is a network link, and*
- ***portion*** *is a capacity portion of* ***link*** *for* ***slot****,*

- buying bids *of the form* $b_y = \left(\boldsymbol{id}, \bigwedge_{\boldsymbol{link}} \bigvee_{\boldsymbol{portion}} (\boldsymbol{slot}, \boldsymbol{link}, \boldsymbol{portion}), p\right)$*, where*
  - ***id*** *is the buyer's identity,*
  - ***slot****,* ***link*** *and* ***portion*** *are as above, and*
  - p *is the bid amount,*

and a combinatorial exchange market AND-MULT with

- selling bids *of the form* $b_s = (\boldsymbol{slot}, \boldsymbol{link}, \boldsymbol{q})$ *where*
  - ***slot*** *is a time slot,*
  - ***link*** *is a network link, and*
  - ***q*** *is a quantity (that is, a positive integer),*
- buying bids *of the form* $b_y = (\boldsymbol{id}, \bigwedge_{\boldsymbol{link}} (\boldsymbol{slot}, \boldsymbol{link}), p)$*, where the variable's interpretations are as above.*

Then there are mappings T_b resp. T_s that map any any buying bid b_y of AND-MULT to a set of buying bids $T_b(b_y)$ of AND-OR, and any selling bid b_s of AND-MULT to a set of selling bids $T_s(b_s)$ of AND-MULT, such that for any bid acceptance function $\chi_{\text{AND-MULT}}$ for AND-MULT, there is a bid acceptance function $\chi_{\text{AND-OR}}$ for AND-OR, such that for any given sets of buying and selling bids for AND-MULT, B_y and B_s, a buying bid $b_y \in B_y$ is accepted by $\chi_{\text{AND-MULT}}$ if and only if all buying bids in $T_b(b_y)$ are accepted by $\chi_{\text{AND-OR}}$, and similarly, selling bids $b_s \in B_s$ are accepted by $\chi_{\text{AND-MULT}}$ if and only if all selling bids in $T_s(b_s)$ are accepted by $\chi_{\text{AND-OR}}$.

Proof. Define T_s and T_b as

$$b_s = (\mathbf{slot}, \mathbf{link}, \mathbf{q}) \mapsto T_s(b_s) = \{(\mathbf{slot}, \mathbf{link}, \mathbf{portion}) : \mathbf{portion} = 1, \ldots, q\} \tag{4.1}$$

$$b_y = \left(\mathbf{id}, \bigwedge_{\mathbf{link}} (\mathbf{slot}, \mathbf{link}), p\right) \mapsto T_b(b_y) = \left\{\left(\mathbf{id}, \bigwedge_{\mathbf{link}} \bigvee_{\mathbf{portion}=1}^{n} (\mathbf{slot}, \mathbf{link}, \mathbf{portion}), p\right)\right\} \tag{4.2}$$

Define $\chi_{\text{AND-OR}}$ as

$$\chi_{\text{AND-OR}}(b_y^{\text{AND-OR}}) = 1 \tag{4.3}$$

iff

$$b_y^{\text{AND-OR}} \in T_b(b_y^{\text{AND-MULT}}) \text{ and } \chi_{\text{AND-MULT}}(b_y^{\text{AND-MULT}}) = 1. \tag{4.4}$$

We have to show that if $\chi_{\text{AND-MULT}}$ is a valid bid acceptance function, then so is $\chi_{\text{AND-OR}}$. The first means that if for all slots $\mathbf{slot}_0$ and links $\mathbf{link}_0$ and selling bids $b_s = (\mathbf{slot}, \mathbf{link}, \mathbf{q})$,

$$\left|\left\{b_y = \left((\mathbf{slot}_0, \mathbf{link}_0) \wedge \bigwedge_{\mathbf{link}} (\mathbf{slot}_0, \mathbf{link})\right) : \chi_{\text{AND-MULT}}(b_y) = 1\right\}\right| \leq q. \tag{4.5}$$

The latter means that there is an allocation function ϕ that allocates triples

$$(\mathbf{slot}, \mathbf{link}, \mathbf{portion}) \tag{4.6}$$

to bidders such that every triple is allocated at most once, and that if

$$\chi_{\text{AND-OR}}\left(\left(\mathbf{id}, \bigwedge_{\mathbf{link}} \bigvee_{\mathbf{portion}} (\mathbf{slot}, \mathbf{link}, \mathbf{portion}), p\right)\right) = 1, \tag{4.7}$$

then there is a triple $(\mathbf{slot}, \mathbf{link}, \mathbf{portion})$ with $\phi((\mathbf{slot}, \mathbf{link}, \mathbf{portion})) = \mathbf{id}$. But clearly, equations (4.1) and (4.5) imply such a ϕ exists. $\square$

If two selling bids offer portions in the same spot and are of the same size, we can safely assume that buyers are indifferent between the two bids. This means that rational buyers that wish to acquire one portion for a certain slot will always submit bids that OR-connect all selling bids for the same slot. But this means that the additional expressive power that AND-OR provides more than AND-MULT, is not being used by the bidders. Consequently, *a market that sells link capacities as multiple copies of identical goods will always be preferred over a market that sells link capacity portions as individual goods*. We are therefore interested in the extension of SBNL that admits selling bids with multiple copies of the same good.

4.1 Background: The RSVP protocol

Reservation protocols for network resources, most prominently bandwidth, have been suggested for a long time now. While many applications, like file transfer or email, rarely create short-term bandwidth shortages, multimedia applications like video streaming are more demanding. If both *real time transmission* and *high bandwidth* are required, even generously designed networks quickly run into temporary capacity deficits.

One of the most popular resource reservation protocols is the *Resource Reservation Setup Protocol (RSVP)* ([10], see [79]) which was designed with its application to multicast videoconferencing in mind. RSVP uses *PATH* messages that are sent by the stream source to the potential receivers, and *RESV* messages that travel the opposite way and carry reservation requests from the receiver to the source. Reservation requests are processed hop-by-hop by the routers which are responsible for acceptance or rejection of reservations. A rejected reservation is not propagated further upstream, and an error messages is sent in reply to the issuer of the request. Two aspects of RSVP are of interest here:

- RSVP reservation messages can either request *controlled load* service, or *guaranteed* service. Controlled load requests are specified by *traffic specifications (TSpecs)* which contain parameters that describe the anticipated traffic, like average and peak rate, package size, etc. Guaranteed service is characterized by *service rate* (the bandwidth in bytes/second), and *slack* (the additional delay that the hop may add, in microseconds).

- Reservation messages are for *immediate* resource usage, there is no advanced reservation for time slots in the future.

RSVP does not specify how RESV messages are processed. Clearly, the protocol was designed with the intention that bandwidth is reserved at a first-come first-served base. It is, however, conceivable that monetary bids are added to RESV messages, and that reservation requests are aggregated at intermediate hops in order to implement an allocation based on monetary bids. Reservations in RSVP expire unless renewed within a given time (specified in the TIME_VALUE field), but if they are renewed, they remain valid.

If bid amounts refer to a reservation whose duration is indefinite, any bid acceptance mechanism obviously leads to inefficiencies of arbitrary degree. There are only two ways out of that: either reservations are for a bounded time, or reservations are unbounded but it is accepted that possibly, flows will be interrupted. Among others, Burchard [12] uses the first approach: reservations apply to certain time slots well-known in advance. Extensions to RSVP that support reservations for given time slots have been suggested e.g. in [69]. While indefinite reservations are not considered in [69], it is possible to re-negotiate the duration of reservations, albeit without a guarantee of success. An overview about advance resource reservation is given in [80], but in respect to reservation with unknown in advance duration, the authors note only that they are "difficult to implement".

4.2 An auction market for advanced reservations with well-known duration

In this scenario, we have

- *selling bids* of the type $b_s = (\mathbf{slot}, \mathbf{link}, \mathbf{q})$, and
- *buying bids* of type

$$b_y = \left(\mathbf{id}, \bigwedge_{\mathbf{slot} \in \mathbf{slotset}} \bigwedge_{\mathbf{link} \in \mathbf{linkset}} (\mathbf{slot}, \mathbf{link}), p\right), \tag{4.8}$$

  where **slotset** is the set of **slot**s that define the time period that the reservation request refers to[2], and **linkset** is the set of **link**s that flow travels through.

It is easy to see that this yields an instance of the market from definition 52; one just needs some enumeration of $\{(\mathbf{slot}, \mathbf{link}) : \mathbf{slot} \in \mathbf{slotset}, \mathbf{link} \in \mathbf{linkset}\}$ that maps slot-link pairs of b_y to the corresponding is in $(k_1^{b_y}, \ldots, k_i^{b_y}, \ldots, k_n^{b_y}, p)$ of equation (3.21).

Let

$$B_y = \left\{ b_y = \left(\mathbf{id}, \bigwedge_{\mathbf{slot} \in \mathbf{slotset}} \bigwedge_{\mathbf{link} \in \mathbf{linkset}} (\mathbf{slot}, \mathbf{link}), p\right) : b_y \in B_y \right\} \tag{4.9}$$

be a set of reservation request bids. For a slot-link-pair $\mathbf{sl} = (\mathbf{slot}, \mathbf{link})$, let $B_y^{\mathbf{sl}}$ be the set of all bids in B_y that bid for $\mathbf{sl}$. Condition 3.22, applied to our market, then says for any valid bid acceptance function χ, for any slot-link-pair $\mathbf{sl} = (\mathbf{slot}, \mathbf{link})$, the condition

$$\sum_{b_y \in B_y^{\mathbf{sl}}} \chi(b_y) \leq 1 \tag{4.10}$$

holds.

How is the matching of buying and selling bids to be timed? An advantage of commit window clearing (see definition 48) is that auctions don't have to be synchronized exactly, it is enough if the time windows $(t_{\text{earliest}}, t_{\text{latest}})$ where a commit is possible do overlap. For a slot $\mathbf{slot} = (t_{\text{start}}^{\mathbf{slot}}, t_{\text{end}}^{\mathbf{slot}})$, clearly we should have $t_{\text{latest}} \leq t_{\text{start}}^{\mathbf{slot}}$, and there is no need for a stricter condition. But what should be the earliest time that a bid is accepted? If uniformly for all slots, the slot length is $l^{\mathbf{slot}}$ and $\Delta^{\mathbf{slot}}$ is such that we allow commit for a slot $\mathbf{slot} = (t_{\text{start}}^{\mathbf{slot}}, t_{\text{end}}^{\mathbf{slot}})$ from time

[2] We do not formally require **slotset** to be a contiguous set of time slots.

$t_{\text{start}}^{\textbf{slot}} - \Delta^{\textbf{slot}}$, for reservation request that requests resources between $t_{\text{start}}^{\text{res}}$ and $t_{\text{end}}^{\text{res}}$, to be granted it is necessary that

$$t_{\text{end}}^{\text{res}} \leq t_{\text{start}}^{\text{res}} + \Delta^{\textbf{slot}} + l^{\textbf{slot}} \tag{4.11}$$

or

$$\text{span}^{\text{res}} \leq \Delta^{\textbf{slot}} + l^{\textbf{slot}} \tag{4.12}$$

with $\text{span}^{\text{res}} = t_{\text{end}}^{\text{res}} - t_{\text{start}}^{\text{res}}$ is the reservation's time span.

4.3 Unknown reservation duration: Extensions to the admission control algorithm of Greenberg et al. for a single link

If reservation duration is unknown (and with no known bounds) but accepted reservations *must* be honoured, then a link that has accepted a reservation for a slot **slot** can't accept reservations for any slot after **slot** until the given reservation has been released.

Greenberg et al. [31] [3] proposes the parallel use of *book-ahead* requests (BA) with known duration, and *instantaneous requests* (IR) which are of unlimited validity. While book-ahead requests are guaranteed once accepted, it is admissible that service based on instantaneous requests is interrupted or downgraded with a sufficiently small probability. Greenberg et al. assume that BA and IR requests arrivals are given as independent Poisson processes with rates λ_B, λ_I and that BA and IR holding times (known in advance for BA and unknown for IR) are independent and exponentially distributed with means $\frac{1}{\mu_B}, \frac{1}{\mu_I}$. Furthermore it is assumed that total link capacity is s, that IR requests are for capacity 1 and BA requests are (uniformly) for capacity b. Let r_B and r_I be the per time unit rates paid for completed BA and IR requests, and let C_I be the "penalty" (cost) for an interrupted IR request[4]. Then there are three variables that control the total generated revenue:

- the probability P_I that an incoming IR request is rejected,
- the probability P_B that an incoming BA request is rejected, and
- the probability p_I that an admitted IR request is interrupted while in progress.

[3] A similar approach was presented in [68].

[4] There is no penalty if the request was rejected or if the reservation was released by the user.

Parameter	Definition	Known to network	Known to user	Depends on
s	link capacity	yes	no	
r	link capacity put aside for IR requests	yes	no	s, b, λ_B, λ_I, μ_I, μ_B
b	capacity of BA requests	yes	yes	
λ_B	BA arrival rate	yes	no	
λ_I	IR arrival rate	yes	no	
μ_I	IR holding parameter	yes	yes	
μ_B	BA holding parameter	yes	yes	
p_I^{emp}	empirical interruption probability	yes	yes	s, b, λ_B, λ_I, μ_I, μ_B
p_I	interruption probability with known existing IR and BA reservations	yes	no	existing IR and BA reservations
p_I^{max}	maximal interruption probability such that IR request is accepted	yes	no	s, b, λ_B, λ_I, μ_I, μ_B
$\mu_{\text{int}}^{\text{emp}}$	$P(t_{\text{int}} < t) = 1 - e^{-\mu_{\text{int}}^{\text{emp}} t}$, where t_{int} is the time when a (never released) reservation is interrupted	yes	yes	p_I^{emp}, μ_I
r_a	per-time utility of user a	no	yes	
C_a	interruption cost for user a	no	yes	
r_p	payable rate for some reservation	yes	yes	p_I
C_p	compensation for interrupting some reservation	yes	yes	p_I
t_x	time when user stops usage of resource reservation	no	yes	
t'_x	time when resource reservation is released	yes	yes	$r_p, C_p, \mu_{\text{int}}$

Table 4.2: Parameters in the Greenberg et al. model

With these variables, we can write the revenue rate per time unit

$$R = (1 - P_I)\,(1 - p_I)\,\frac{r_I}{\mu_I'}\lambda_I \,+\, (1 - P_B)\,\frac{r_B}{\mu_B}\lambda_B \,-\, p_I\,(1 - P_I)\,C_I\lambda_I \tag{4.13}$$

Here μ_I' is the mean holding time for IR request *conditioned upon that the request is not interrupted*. If p_I is small, then we can approximate μ_I' by μ_I.

- With the assumption that BA requests are far into the future, the admission control for BA requests can take place without consideration of the IR calls in progress. A BA request will be admitted if, including that request, at no time more that $s - r$ capacity is reserved by BA reservation. The parameter r defines an amount of capacity that is “put aside” for IR calls.

- On the other hand, the admission control for IR request must take into consideration the IR calls in progress as well as the pending BA reservation. From that information, the probability p_I is computed, and the IR request is granted if and only if the interruption probability for this request is less than $p_I^{\max}$, with $p_I^{\max}$ being the “threshold probability” parameter.

Thus, the variables P_I, P_B and p_I that control the revenue all depend on the parameters r and $p_I^{\max}$. The maximization of R in (4.13) takes place by variation of r and $p_I^{\max}$.

Greenberg et al. sketch how p_I can be computed and point out that the computation is somewhat cumbersome. They suggest different approximations whose precision they evaluate in simulations. Greenberg et al.’s simulations also showed that allowing BA and IR requests can significantly increase generated revenue even if the probability threshold for service interruption is small.

4.3.1 Bidding for paths in the admission protocol of Greenberg et al.

Greenberg et al.’s admission control mechanism can easily be extended to the case that a reservation request asks for a combination of resources: the admission control algorithm is run separately for every link, and the request is granted if and only if *all* links are available. Figure 4.1 shows a summary of the extended algorithm.

The procedures `AdmitBA` and `AdmitIR` are called when requests arrive. They use the global variables `AdmittedBA` and `ASdmittedIR` whose keeping up-to-date we have omitted in the pseudocode. The procedure `InterruptProbability` performs the (approximate) computation of p_I as in [31]. The parameters $p_I^{\max}$ and r are either constants fixed in advance, or could also be dynamically adopted to maximize R in (4.13).

4.3.2 Mechanism design for reservations with unknown duration.

Greenberg et al. assume that the pay rates r_I and r_B and the “penalty” for interrupted IR reservations C_I are constants for all requests (which than can be set to 1

```
globals AdmittedBA(link) // set of admitted BAs per link
        AdmittedIR(link) // set of IR calls in progress per link
procedure AdmitBAperLink(slotset,link)
  forall (slot in slotset) do
    n=0
    forall ( (islotset,ilinkset) in AdmittedBA(link) ) do
      if (slot is in islotset)
        n++
      endif
    end
    if (n*b>c − r)
      return false
    endif
  end
  return true
end procedure

procedure AdmitIRperLink(link)
  pir=InterruptProbability(AdmittedIR(link),AdmittedBA(link))
  if (pir < p_I^max)
    return true
  else
    return false
  endif
end procedure

procedure AdmitBA(slotset,linkset)
  forall (link in linkset) do
    if (AdmitBAperLink(slotset,link) == false)
      return false
    endif
  end
  return true
end procedure

procedure AdmitIR(linkset)
  forall (link in linkset) do
    if (AdmitIRperLink(link) == false)
      return false
    endif
  end
  return true
end procedure
```

Figure 4.1: Admission control for combinatorial requests.

without loss of generality).

We now consider a market where users submit buying bids

$$b_y = \left(\mathbf{id}^{b_y}, \bigwedge_{\mathbf{slot} \in \mathbf{slotset}^{b_y}} \bigwedge_{\mathbf{link} \in \mathbf{linkset}^{b_y}} (\mathbf{slot}, \mathbf{link}), r^{b_y}, C^{b_y} \right) \tag{4.14}$$

and selling bids $b_s = (\mathbf{slot}, \mathbf{link}, \mathbf{q})$.

How would a good mechanism for this market look like?

First note that if the per-time rate is payable only for completed calls, there is no incentive for a user ever to finish a call. Therefore, this approach is unfeasible. We thus modify the revenue R of (4.13) such that the rate r_I is payable even for interrupted calls[5]. This yields

$$R = (1 - P_I)\, \frac{r_I}{\mu_I} \lambda_I + (1 - P_B)\, \frac{r_B}{\mu_B} \lambda_B - p_I\, (1 - P_I)\, C_I \lambda_I \tag{4.15}$$

A first glance may suggest a mechanism that pays a (per-time usage) price payable if the request is accepted, and a “penalty” that is paid to the bidder in case that the request is accepted but the service is interrupted. The problem with that approach is that if the duration of the request is controlled by the bidder, he may manipulate by intentionally not terminating the service to be entitled to the penalty payment.

So a user a’s *type* is a tuple (r_a, C_a) where r_a is the *utility rate* and C_a is the *cost of interrupt*. Furthermore, we assume that the *duration* of a reservation is a random variable t_x with exponential distribution. The value of t_x , is unknown even to a itself at the time when he places his bid. We assume that after t_x has expired, the user has no further utility from his reservation. However, he can keep up the reservation to speculate on the interruption penalty C_I. The utility of a thus depends on a parameter t'_x that a chooses and that defines the time *that a releases his reservation, provided that it hasn’t been cancelled before*. This decision has to be made only after t_x is known, so t'_x can be dependent on t_x. a will choose t'_x in such a way that his expected utility is maximized.

Let is compute the expected utility that a gains from the call *after t_x has expired*, conditioned on t_x and the assumption that the call was not interrupted so far. Let t_{int} be the time when the reservation is interrupted. Greenberg et al. give a computation of the interruption probability for a call, depending on the currently existing BA reservation. While the reservation owner does indeed have this information, a does not. Therefore, it is safe for a to assume that the interruption probability time for any given call (of infinite duration) is also exponentially distributed with parameter

[5] One might argue that this is quite a realistic model anyway: the “interruption cost” C_I could then be interpreted as the cost of the inconvenience to re-build the connection, while the utility already gained by the service before the interruption is not lost.

μ_{int}. Then

$$u_a(t'_x|t_x, t_{\text{int}} > t_x) = \left(\int_{t_x}^{t'_x} \mu_{\text{int}} e^{-\mu_{\text{int}}(t-t_x)} \left(C_p - (t-t_x) r_p\right) \mathrm{d}t \right)$$
$$- \left(t'_x - t_x\right) r_p P(t_{\text{int}} > t'_x) \tag{4.16}$$
$$= \frac{e^{t'_x \mu_{\text{int}}} \left(e^{t'_x \mu_{\text{int}}} - e^{t_x \mu_{\text{int}}}\right) \left(C_p \mu_{\text{int}} - r_p\right)}{\mu_{\text{int}}} \tag{4.17}$$
$$\longrightarrow_{t'_x \to \infty} C - \frac{r_p}{\mu_{\text{int}}} \tag{4.18}$$

with r_p be the *price rate* that a has to pay and C_p is the amount of the interruption compensation payment.

So a has an expected win from choosing $t'_x > t_x$ if and only if $\mu_{\text{int}} C_p > r_p$.

How can a get an estimate of μ_{int}?

Case 1. We assume that the network attempts to optimize welfare and *drops least valuable calls first*. In this case, some information is required on the distribution of utility rates and interruption costs. We don't consider this case here.

Case 2. We follow Greenberg et al. and let the network *interrupt younger calls first*. Under this assumption, a can compute $\mu_{\text{int}}^{\text{emp}}$ from the empirical interruption probability p_I^{emp} and the parameter μ_I of the (exponentially distributed) *call duration distribution*:

$$p_I^{\text{emp}} = \int_0^\infty P(t_{\text{int}} < t_{\text{dur}}) f(t_{\text{dur}}) \mathrm{d}t_{\text{dur}} \tag{4.19}$$
$$= \int_0^\infty \left(1 - e^{-\mu_{\text{int}} t_{\text{dur}}}\right) \mu_I e^{-\mu_I t_{\text{dur}}} \mathrm{d}t_{\text{dur}} \tag{4.20}$$
$$= \mu_I \left(\frac{1}{\mu_I} - \frac{1}{\mu_{\text{int}} + \mu_I} \right) \tag{4.21}$$
$$= 1 - \frac{\mu_I}{\mu_{\text{int}} + \mu_I} \tag{4.22}$$

and therefore

$$\mu_{\text{int}} = \mu_I \frac{p_I^{\text{emp}}}{1 - p_I^{\text{emp}}}. \tag{4.23}$$

Note that for a given pair (r_p, C_p), both a and the network can compute μ_{int}, and therefore know whether "speculating" on C_p is profitable and also the amount of the expected profit.

4.3.2.1 Non-existence of efficient two-dimensional mechanisms.

Definition 55. *A* two-dimensional access control mechanism *takes as input bids of the form* $(b_a = r_a, C_a)$ *and computes a decision function* χ *with* $\chi(b_a) \in \{0, 1\}$*, and a* rate-compensation vector (r_p, C_p) *such that* r_p *is the rate that* a *pays per time unit until the reservation is released by* a *or cancelled by the network, and* C_p *is the compensation that* a *receives if the reservation is interrupted by the network.*

We say that a mechanism satisfies rate-compensation voluntary participation *if always* $r_p \leq r_a$ *and* $C_p \geq C_a$*.*

Corollary 56. *If the interruption probability* p_I^{emp} *is publicly known, there is no efficient two-dimensional access control mechanism.*

Proof. Suppose there is only one bid (r_a, C_a) with $\mu_{\text{int}} C_a > p_a$.

If the mechanism is efficient, it must accept the single bid. Let (r_m, C_m) be the rate-compensation vector returned by the mechanism.

Case 1: $r_m \leq r_a$ *and* $C_m \geq C_a$*.* Then speculating on C_m is profitable for a. But then the mechanism can't be efficient.

Case 2: $r_m > r_a$ *or* $C_m < C_a$*.* Without loss of generality we can assume that a has revealed his true type. But then it would be profitable for release the reservation right away. But this contradicts efficiency, because the request is lost even if capacity is not used up. □

4.3.2.2 Mapping two-dimensional types to one-dimensional ones.

In order to apply standard VGC mechanisms, we have to project the two-dimensional types and bids (r, C) to one-dimensional ones. Here a one-dimensional bid is a per-time price r, and no compensation being paid for service interrupts. The projection π has to satisfy that a risk-neutral user a is indifferent between the choices of either being offered rate r and a compensation C if service is interrupted, and a ("discounted") rate $\pi(r, C)$ and no interrupt compensation. This is the case when the expected profit from (r, C) and from $\pi(r, C)$ are equal. The fact that the only difference between the two options is in payment implies that expected payments must be equal.

While the *empirical* interrupt probability p_I^{emp} (and the distribution $f_{\text{int}}^{\text{emp}}$ and the parameter $\mu_{\text{int}}^{\text{emp}}$ of the corresponding exponential distribution) is public information (see table 4.2), the network additionally knows at any given time the currently present BA reservations and can use this additional information to compute the interrupt distribution function. Note that even if the expected profit computation is based on the currently present BA reservations, it is obvious that there always *is* some projection π such that both expected profits are equal.

The use of the approximated distribution function f_{int} yields

$$E(\text{payment}(r,C)) = \int_0^\infty \mu_I e^{-\mu_I t_x} \left[\left(\int_0^{t_x} f_{\text{int}}^{\text{emp}}(t'_x)(-rt'_x + C)\mathrm{d}t'_x \right) \right.$$
$$\left. - P(t_{\text{int}} > t_x) r t_x \right] \mathrm{d}t_x \tag{4.24}$$
$$= \int_0^\infty \mu_I e^{-\mu_I t_x} \left[\left(\int_0^{t_x} \mu_{\text{int}}^{\text{emp}} e^{-\mu_{\text{int}}^{\text{emp}} t'_x}(-rt'_x + C)\mathrm{d}t'_x \right) \right.$$
$$\left. - e^{-\mu_{\text{int}}^{\text{emp}} t_x} r t_x \right] \mathrm{d}t_x \tag{4.25}$$
$$= \frac{-r + C\mu_{\text{int}}^{\text{emp}}}{\mu_I + \mu_{\text{int}}^{\text{emp}}} \tag{4.26}$$

and

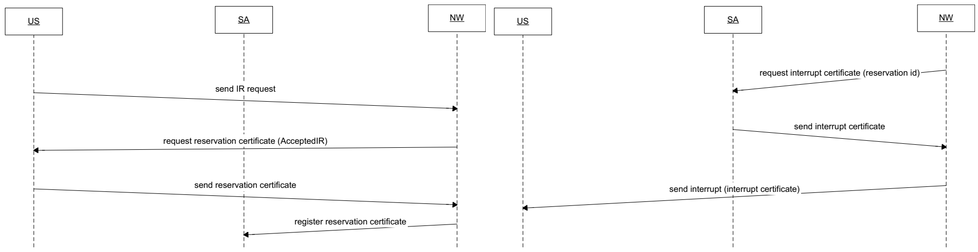
$$E(\text{payment}(\pi(r,C))) = \int_0^\infty \mu_I e^{-\mu_I t_x} \left[\left(\int_0^{t_x} -f_{\text{int}}^{\text{emp}}(t'_x) t'_x \pi(r,C)\mathrm{d}t'_x \right) \right.$$
$$\left. - P(t_{\text{int}} > t_x) \pi(r,C) t_x \right] \mathrm{d}t_x \tag{4.27}$$
$$= \int_0^\infty \mu_I e^{-\mu_I t_x} \left[\left(\int_0^{t_x} -\mu_{\text{int}}^{\text{emp}} e^{-\mu_{\text{int}}^{\text{emp}} t'_x} t'_x \pi(r,C)\mathrm{d}t'_x \right) \right.$$
$$\left. - e^{-\mu_{\text{int}}^{\text{emp}} t_x} \pi(r,C) t_x \right] \mathrm{d}t_x \tag{4.28}$$
$$= -\frac{\pi(r,C)}{\mu_I + \mu_{\text{int}}^{\text{emp}}} \tag{4.29}$$

Simplifying yields

$$\pi(r,C) = r - C\mu_{\text{int}}^{\text{emp}} \tag{4.30}$$

4.3.2.3 The empirical interrupt probability.

The above assumed that users have access to an empirical interrupt probability p_I^{emp}. This probability is public information, and is equal for all users. It is not necessary that p_I^{emp} be a constant over time, that is, it can be time-dependent. For instance, it is conceivable that p_I^{emp} is drawn from previous days and depending on the time of the day.



(a) Verified counting of reservation requests (b) Verified counting of service interrupts

Figure 4.2: Verified interrupt statistics

An important point is that, using standard cryptographic infrastructure, p_I^{emp} can be *independently verified* to guarantee its accuracy even if mutual trust between users and the network is absent. The verification algorithm uses that

- it is in the interest of the *network* to have a low published interruption probability (because then, users will bid higher according to equation (4.30)), while
- the *users* want certainty that p_I^{emp} is not to optimistic, because they incur losses with every interruption.

The verification assumes that there is a trusted by all sides *statistics authority* `SA` that counts admitted reservation requests and interrupts, and publishes p_I^{emp}. The data collection used by `SA` is built into the protocols for IR reservation requests and IR interrupts. Let `NW` denote the network, and `US` the user. `NW` maintains a counter `AcceptedIR`. If `NW` accepts an IR request, it increments `AcceptedIR` and sends its value as a challenge to `US`. `US` replies with a *reservation certificate* that contains the signed value. `NW` registers the reservation certificate at `SA`.

In case of a service interruption, `NW` requests from `SA` a *interrupt certificate* containing some identification of the IR to be interrupted. `NW` sends the certificate to the affected `US`. `SA` counts every reservation certificate as an admitted reservation (not accepting duplicates with identical counter `AcceptedIR`), and every interrupt certificate as a service interrupt. `NW` won't serve IR reservations without the user issuing a proper reservation certificate. Users complain if service interruption occurs without `NW` presenting a corresponding interrupt certificate.

Figure 4.2 shows an overview of the verified counting protocol.

4.4 Is there an advantage in auctioning bandwidth ?

We have above demonstrated how auction solutions can be used for bandwidth allocation. So far, we have not stated how such a "market solution" performs in comparison with the classical *first come first served* allocation with a fixed rate.

For the comparison, we assume that there is a single link of fixed capacity c which is allocated to requests issued by users in a random (Poisson) process with fixed parameter λ. Granted requests are served until they are terminated after an exponentially with parameter d distributed time t, and users are charged rt where r is a *rate* determined by the allocation mechanism. While in the fixed price scenario, the rate is a constant at all times, the auction mechanism recomputes the rate periodically on the base of the currently issued requests.

Requests have an associated *maximum rate* the issuer is willing to pay, and it is guaranteed that a request is never charged more than this maximum rate. We assume that the maximum rate of any request is drawn from a normal distribution with constant parameters μ and σ. The fixed price mechanism always denies all requests with a maximum rate lower than the fixed rate. The auctioning mechanism computes a *current rate* and accepts all requests with maximum rate at least the current rate, and denies all other requests. The mechanism guarantees the current rate for the complete time until the request is terminated by the issuer, no matter how future current rates develop.

We simplify our scenario by assuming that there are no BA calls, and therefore, there is no need of cancelling requests from the side of the mechanism.

In order to perform an auction, the mechanism needs to collect requests over a certain time span which without loss of generality be of length 1.

4.4.1 Fixed price mechanism

Let us analyze the mechanism that offers a fixed rate r_f. Requests are generated as a Poisson process with parameter λ, and the associated maximum rates are normally distributed with parameters μ and σ. As described above, requests with associated rate less than r are discarded. Thus, the sequence of non-discarded requests is generated by a Poisson process with parameters $\lambda^*(r_f) = \lambda P(r > r_f)$ where $r \sim N(\mu, \sigma)$. Write $f(r)$ and $F(r)$ for the probability density function and cumulative probability function of $N(\mu, \sigma)$. So

$$\lambda^*(r_f) = \lambda \left(1 - \frac{1}{2}\left(1 + \mathrm{Erf}\left(\frac{r_f - \mu}{\sqrt{2}\sigma}\right)\right)\right) \tag{4.31}$$

with Erf being the well-known *error function* $\mathrm{Erf}(z) = \frac{2}{\sqrt{\pi}} \int_0^z e^{-z^2}\, dz$.

Since the requests of the modified process are served on first-come first-served base, the reservation state for the fixed price scenario can be modelled as a queue with Poisson arrival, exponential living time, c servers and no additional waiting room. It follows (see e.g. [60, p.65]) that the state probability distribution is

$$p_c(k) = \frac{(\frac{\lambda^*}{d})^k}{k!\left(1 + \sum_{l=1}^{c} \frac{(\frac{\lambda^*}{d})^l}{l!}\right)} \tag{4.32}$$

and the mean generated revenue per step can be computed by

$$\text{meanrevenue}(c, r_f) = \sum_{k=1}^{c} (r_f k p_c(k)) \tag{4.33}$$

Miller [51] considers M/M/c/c queues with a discrete set of customer classes. They give an algorithm to compute the optimal admission policy if decision is immediate, that is, the decision depends on the number of customers currently in the queue. They introduce "shadow costs" $\nabla y_i = y_{i+1} - y_i$ of serving a customer if the system is in state i that reflect the expected revenue lost from the fact that now one more server is busy. They show that than the following holds:

$$\begin{array}{llll} \lambda(b - \nabla y_0) & & & = A \\ & \cdots & & \\ \lambda(b - \nabla y_1) & + & i\mu \nabla y_{i-1} & = A \\ & \cdots & & \\ & & c\mu \nabla y_{c-1} & = A \end{array} \tag{4.34}$$

where A is the per-time generated revenue (∇y_i and A depend on the admission policy.)

Miller and Buckman [52] compare revenues under optimal state-dependent admission policies with the ones with the optimally chosen fixed price. They assume that customer utilities are exponentially distributed. They conclude that

> in a more realistic setting where economic environment is uncertain, calculations suggest that there is a greater incentive to use an optimal transfer pricing policy.

Furthermore, Miller and Buckman compute the *optimal value T^* of the fixed price* T, by maximizing A. Their theorem 2 says that for the optimal value T^*, the following holds:

$$T^* = \sum_{i=0}^{c-1} q_i \nabla y_i(c, T^*) \tag{4.35}$$

with

$$q_i = \frac{p_i(c)}{\sum_{j=0}^{c-1} p_j(c)} \tag{4.36}$$

are the steady state probabilities of the queue conditioned on not all servers being busy.

Low [45] computes optimal service price in a M/M/s/c queue if service prices are computed in advanced after a new service has been accepted into the queue, or a service was completed. Low assumes that at any such instance, the price for the next arriving customer is chosen from a set P of possible prices with P being finite or a bounded closed subinterval of $\mathbb{R}$. Low proves the existence of a stationary strategy maximizing average per-time revenue, and gives an algorithm for its computation.

Ziya et al. [85] compute the *revenue-maximizing rate* r_f^{opt}. They prove that (proposition 6.1)

$$r_f^{\text{opt}}(c) = \inf\{r : e(r)\nabla_c(r) \geq 1\} \tag{4.37}$$

where

$$e(r) = r\frac{f(r)}{1-F(r)} \text{ is the } \textit{price elasticity, and} \tag{4.38}$$

$$\nabla_c(r) = 1 + \frac{\lambda^*}{d}(p_c(c) - p_{c-1}(c-1)) \tag{4.39}$$

4.4.2 Vickrey price mechanism

For a fair comparison of the revenue, we assume that the reservation requests are generated exactly as above. However, resource allocation takes place at the end of each time interval of length 1.[6] Let $t_k = (k-1, k]$ be the kth interval. Let $\{r_1^k, \dots, r_{n^k}^k\}$ with $r_1^k \leq \dots r_{n^k}^k$ be the requests submitted during t_k. (So n^k has Poisson distribution with parameter λ.) Let l^k be the number of living requests at time k. Note that l^k is distributed according to the state distribution of a queue with Poisson(λ) arrival and exponential(d) departure with c servers and no waiting room.

The requests r_i^k are treated as bids for an auction of a good that is available in $c - l^k$ copies. So all requests r_i^k with $i \geq n^k - c + l^k + 1$ are accepted and pay the rate

$$r^k = \begin{cases} r_{n^k-c+l^k} & \text{if } n^k - c + l^k \geq 1 \\ 0 & \text{otherwise.} \end{cases} \tag{4.40}$$

The mean revenue per step then is

$$\text{meanrevenue} = cE(r^k) \tag{4.41}$$

4.4.3 Comparing the revenue

In the following, we give a comparison of the revenues of Vickrey pricing and pricing with an optimally chosen fixed price. It will be shown that in many cases, Vickrey pricing generates a higher revenue than optimal fixed pricing. We remark that to compute the optimal fixed price, it is necessary to have a priori assumptions on the distribution of requests. Vickrey pricing, however, generates a high revenue even if there is no information on the distribution parameters of the incoming requests.

[6] This means that clients have to accept a delay < 1 until their reservations are accepted or rejected.

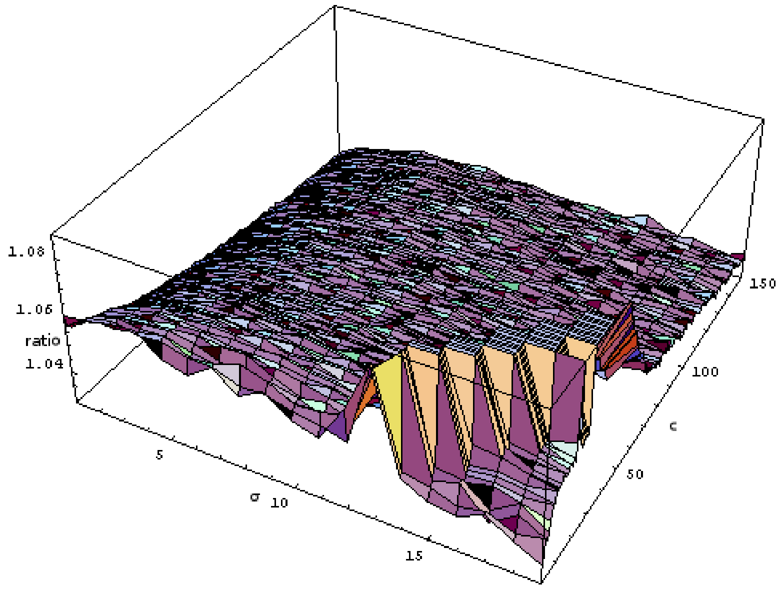


Figure 4.3: Revenue ratio Vickrey vs. fixed pricing with $\lambda = 100, d = 0.1, \mu = 10, \sigma \in [0, 20], c \in [1, 150]$

4.4.3.1 No global inequality.

While we intend to demonstrate that Vickrey pricing generates in some cases more revenue than fixed rate pricing even with the optimally chosen fixed price, there is no global inequality: Indeed, for $\frac{\lambda}{d} \ll c$ and small σ, the Vickrey price will converge to 0 for $c \to \infty$ (since the probability that less than c requests are in the system converges to 1) while the number of accepted bids is bounded by the number of submitted bids, and consequently the Vickrey revenue converges to 0 if the capacity grows beyond all limits, but the fixed rate revenue is still significantly positive. To be more precise, we get from (4.32) and (4.33) for any fixed rate r,

$$\begin{aligned}
\lim_{c \to \infty} \text{meanrevenue}(c, r) &= r \sum_{k=1}^{\infty} (k p_{\infty}(k)) \\
&= r \sum_{k=1}^{\infty} \left(k \frac{(\frac{\lambda^*}{d})^k}{k! e^{\frac{\lambda^*}{d}}} \right) \\
&= r \frac{\lambda^*}{d} > 0
\end{aligned} \tag{4.42}$$

4.4.3.2 Simulation results.

Figure 4.3 shows the ratio of the revenues generated by Vickrey pricing and fixed pricing with the optimally chosen fixed price. Note that in the chosen parameter domain, Vickrey pricing generates a slightly higher revenue compared with optimal fixed pricing. The advantage of Vickrey pricing here grows for small capacities and larger σ.

4.5 Summary

In this chapter, we applied the theory of double-sided combinatorial auctions to advanced reservations in networks. In particular, we used extensions of SBNL that allows multiple item and combinatorial auctions. After describing a mechanism that allows fixed length reservations, we presented various results on the case where reservations are open-ended:

- Efficient mechanisms do not exist if the mechanism pays a compensation for service interrupts.
- We showed how to map bids with per-time rate and penalty for service interrupts, to bids only with per-time rate.

With this mapping, it is possible to apply mechanisms for combinatorial exchanges, in particular, the SBNL pricing and commit window clearing as presented in the previous sections.

Finally, we presented simulation results that support that Vickrey pricing generates, in some cases, a higher revenue than fixed pricing with an optimally chosen price, while not requiring information on the distribution of the requests.

A general characterization of the relationship between Vickrey revenue and optimal fixed price revenue would be highly desirable. We, however, leave this point open for future research.

5 Indirect mechanisms for multicast pricing

The previous chapter presented pricing for unicast streams based on the VGC mechanism. We considered *direct* mechanisms where users would reveal their utilities to the network manager who would compute a resource allocation which maximizes total utility.

Another line of research [41, 42, 40] works with *indirect mechanisms*, describing network flows as Walrasian tatonnement with elastic price and demand. Here, network users adopt a control parameter that controls bandwidth allocation (in our case, this parameter can be interpreted as payment either in monetary terms, or in terms of an accepted delay). We adopt the interpretation that users control bandwidth by monetary payment. In the tatonnement process, the auctioneer splits bandwidth in proportion with the submitted payments. The adoption takes place continuously.

Given a set of resources J, a *route* is a subset $r \subseteq J$. Fix set R of possible routes and define $A_{j,r}$ to be the matrix defined by $A_{j,r} = 1$ if $j \in r$, and 0 otherwise. Suppose that every resource j has a *capacity* $c_j \geq 0$. For a route r, let x_r be the *flow through* r. The vector $x = (x_r : r \in R)$ is called the *total flow*. x is *feasible* if for all resources $j \in J$, we have $\sum_{r:j\in r} x_r \leq c_j$. Let us furthermore assume that to every route r, there is an associated user that has utility $u_r(x_r)$ from r which depends on x_r.

Kelly considers three interconnected optimization problems[1]:

- the *system* tries to maximize aggregated utilities:

  SYSTEM(u, A, c)

$$\max \left(\sum_{r \in R} u_r(x_r) \right) \text{ over } (x_r : r \in R) \tag{5.2}$$

[1] The setting described here is known as *inelastic supply* setting. This refers to the fact that every link has a fixed capacity that is split among users. *Elastic supply* settings, see [40] assume that supply can vary but that there is a cost associated with it. The corresponding SYSTEM problem then has the form

$$\max \left(\sum_{r \in R} u_r(x_r) - C \left(\sum_{r} x_r \right) \right) \text{ over } (x_r : r \in R) \tag{5.1}$$

where $C(x)$ is the *cost* associated with a supply of capacity x.

subject to

$$\sum_{\{r \in R : A_{j,r}=1\}} x_r \le c_j \text{ for all } j \in J \tag{5.3}$$

$$x_r \ge 0 \text{ for all } r \in R \tag{5.4}$$

- the *users* maximize their own profit (that is, utility minus costs) while varying the size of the flow he acquires for a given per-unit price λ:

  $\mathrm{USER}_r(u_r, \lambda)$

  $$\max \left(u_r(x_r) - w_r\right) \text{ over } x_r \tag{5.5}$$

  subject to

  $$x_r \ge 0 \tag{5.6}$$

  $$w_r = x_r \lambda_r. \tag{5.7}$$

- Finally, the *network* maximizes revenue by varying the flow sizes:

  $\mathrm{NETWORK}(A, \lambda, c)$

  $$\max \sum_{r \in R} \lambda_r x_r \text{ over } (x_r : r \in R) \tag{5.8}$$

  subject to

  $$\sum_{\{r \in R : j \in r\}} x_r \le c_j \text{ for all } j \in J \tag{5.9}$$

  $$x_r \ge 0 \text{ for all } r \in R. \tag{5.10}$$

Theorem 1 of [40] interconnects these three optimization problems: If the u_r are differentiable, strictly concave functions, then there is a price-per-unit vector $\lambda = (\lambda_r : r \in R)$ such that the unique solution vector $(x = x_r : r \in R)$ of the USER_r problems simultaneously solves $\mathrm{NETWORK}(A, \lambda, c)$, and this vector also solves $\mathrm{SYSTEM}(u, A, c)$.

If we see users and network as self-interested agents (users maximizing their private surplus, while the network maximizes revenue), we are tempted understand this theorem as the guarantee that there is an equilibrium in the associated game, and that at this equilibrium, the *social surplus* as given by the SYSTEM problem (5.2) is also maximized. Note, however, that λ depends on the user's input. If there is a large number of users, and none of their utility functions is dominating, we can assume, for approximation, that the price vector λ does not depend on the input of a fixed user r. Only in this case, the theorem implies that there is an equilibrium with optimal social surplus.

Note that NETWORK(A, λ, c) is the relaxation of the combinatorial auction problem

AUCTION(A, λ, c)

$$\max \sum_{r \in R} \lambda_r x_r \text{ over } x_r \tag{5.11}$$

subject to

$$\sum_{\{r \in R : j \in r\}} x_r \leq c_j \text{ for all } j \in J \tag{5.12}$$

$$x_r \in \{0, 1\} \text{ for all } r \in R. \tag{5.13}$$

where c_j $(1 \leq j \leq J)$ are *goods* and $\lambda = (\lambda_r : r \in R)$ is the *bid vector* of the auction.

In contrast with the work of the mechanism design school, there are no *costs* considered to be incurred by the transmission. Rather, the price is computed such that

- certain *fairness* conditions are honoured, and
- a balance between supply (available capacity of the required resources) and demand is achieved.

5.1 Linear utilities

In [29, par. 6.2], an example is considered that illustrates how the Nash equilibrium is computed in the case that users *are* aware of the effect of their input on the price vector λ:

Assume that n users have linear utility functions $u_r(x_r) = \alpha_r x_r$ with $\alpha_r > 0$. Furthermore, assume that the network assigns rates to the users in proportion of their willingness to pay w_r. For convenience, write $W_r = \sum_{r' \neq r} w_{r'}$ and $S = \sum_r w_r = w_r + W_r$ for all r. Then the maximization problem presented to the user r is

USER$_r(u_r)$

$$\max \left(u_r \left(\frac{w_r}{w_r + W_r} \right) - w_r \right) \text{ over } w_r \tag{5.14}$$

subject to

$$w_r \geq 0 \tag{5.15}$$

Now

$$u_r\left(\frac{w_r}{w_r+W_r}\right) - w_r = \alpha_r \frac{w_r}{w_r+W_r} - w_r \tag{5.16}$$

and thus

$$\frac{\mathrm{d}}{\mathrm{d}w_r}\left(u_r\left(\frac{w_r}{w_r+W_r}\right) - w_r\right) = \frac{\alpha_r}{w_r+W_r}\left(1 - \frac{w_r}{w_r+W_r}\right) - 1 \tag{5.17}$$

Therefore, for the solutions of (5.14) we have

$$\textit{either } w_r = 0 \text{ and } \frac{\mathrm{d}}{\mathrm{d}w_r}\left(u_r\left(\frac{w_r}{w_r+W_r}\right) - w_r\right) < 0, \tag{5.18}$$

$$\textit{or } \frac{\mathrm{d}}{\mathrm{d}w_r}\left(u_r\left(\frac{w_r}{w_r+W_r}\right) - w_r\right) = 0, \tag{5.19}$$

or equivalently,

$$w_r = S\left(1 - \frac{S}{\alpha_r}\right)^+. \tag{5.20}$$

Summing up (5.20) for all r, we get

$$S = S\sum_r \left(1 - \frac{S}{\alpha_r}\right)^+ \tag{5.21}$$

and consequently,

$$1 = \sum_r \left(1 - \frac{S}{\alpha_r}\right)^+. \tag{5.22}$$

Fact 57. *a) There is a unique vector* (w_r) *such that equations (5.20) hold for all* r*, and* $S = \sum_r w_r$*.*

b) For $n = 2$*, we have* $w_r > 0$ *for all* r*.*

Proof. Note first that there is a unique S such that

$$1 = \sum_r \left(1 - \frac{S}{\alpha_r}\right) \tag{5.23}$$

holds for all r, namely

$$S = \frac{n-1}{\sum_r \frac{1}{\alpha_r}}. \tag{5.24}$$

Define

$$v_{r_0} = \frac{n-1}{\left(\sum_r \frac{1}{\alpha_r}\right)^2}\left(\sum_r \frac{1}{\alpha_r} - \frac{n-1}{\alpha_{r_0}}\right) \tag{5.25}$$

which is equivalent to

$$v_{r_0} = S\left(1 - \frac{S}{\alpha_{r_0}}\right). \tag{5.26}$$

Consequently, if $v_r \geq 0$ holds for all r, then with $w_r = v_r$, equations (5.20) and (5.22) hold for all r. Otherwise, remove all r with $v_r < 0$, and apply (5.24-5.25) iteratively. Note that for the case $n = 2$, (5.25) implies that $v_{r_0} > 0$ and thus iteration terminates. This proves existence of S, and also part b of the claim.

To prove uniqueness, assume that there are $S \neq S'$ and $(w_r), (w'_r)$ such that (5.20) holds for all r. Clearly if $w_r > 0$ and $w'_r > 0$, then $w_r = w'_r$. Let r_0 be such that without loss of generality, $w_{r_0} < w'_{r_0}$. Then $w_{r_0} = 0$, $w'_{r_0} > 0$ and thus $S' < \alpha_{r_0} < S$. It follows that for *all* users r, we have $w_r \leq w'_r$. But this implies $S \leq S'$, a contradiction. □

Those users r with $\alpha_r > S$ set $w_r = S\left(1 - \frac{S}{\alpha_r}\right)$ with S according to (5.22), the remaining ones set $w_r = 0$.

5.1.1 Comparison with VGC mechanisms

Note that the equilibrium bid w_r depends on the bids of the other players, and so there is no *dominant* strategy for any user with positive utility. Also, at the equilibrium, the social surplus is *not* maximized. Maximizing social surplus in the case of linear utility functions would mean to allocate *all* capacity to the user r with the highest α_r. This, however, seems absurd: the assumption of linear utility functions is reasonable only as an approximation for small ranges. The tatonnement works independently from the shape of the utility functions, and the results apply when utilities are linear for bandwidth range below the equilibrium allocation.

A dominant strategy mechanism would maximize surplus in respect to *all* possible allocations and would therefore need the complete utility functions (for the bandwidth ranges below the *total capacity*). The applicable VGC mechanism would then be the auction of a divisible good, see the background chapter at 2.4. A dominant strategy mechanism will split resources discontinuously in respect to the input from the clients. This, of course, could impose problems for applications.

5.1.2 Approximativity of the Nash equilibrium

As noted above, the resource allocation at the Nash equilibrium is suboptimal. In this paragraph, we give a bound for the quotient of the utilities of the Nash and optimal allocations.

Note first that the optimal allocation assigns all utility to the users with the highest α_r (being indifferent of how these users share the resource among each other). Utility is then

$$u_{\text{opt}} = \max_r \alpha_r \tag{5.27}$$

For the utility at the Nash equilibrium, we get from (5.24) and (5.25) and with $n = |\{r : w_r > 0\}|$,

$$u_{\text{Nash}} = \sum_{r:w_r>0} \alpha_r \frac{w_r}{S} \tag{5.28}$$

$$= \sum_{r:w_r>0} \alpha_r - \frac{n(n-1)}{\sum_{r:w_r>0} \frac{1}{\alpha_r}} \tag{5.29}$$

and

$$\frac{u_{\text{Nash}}}{u_{\text{opt}}} = \max_{r'} \frac{\sum_{r:w_r>0} \alpha_r - \frac{n(n-1)}{\sum_{r:w_r>0} \frac{1}{\alpha_r}}}{\alpha_{r'}} \tag{5.30}$$

Let $r_{\max} = \arg\max_r \alpha_r$ and define

$$f(\langle \alpha_r : w_r > 0 \rangle) = \frac{\sum_{r:w_r>0} \alpha_r - \frac{n(n-1)}{\sum_{r:w_r>0} \frac{1}{\alpha_r}}}{\alpha_{r_{\max}}} \tag{5.31}$$

This function is symmetric in all α_r except for $r = r_{\max}$. Therefore, at its local minima, $\alpha_r = \alpha$ holds for some α and for all $r \neq r_{\max}$. Now substitute α for α_r with $r \neq r_{\max}$ to receive

$$f(\langle \alpha, \ldots, \alpha, \alpha_r, \alpha, \ldots \rangle) = 1 + (n-1)\alpha \left(\frac{1}{\alpha_{\max}} - \frac{n}{(n-1)\alpha_{\max} + \alpha} \right) \tag{5.32}$$

and for the derivative, we get

$$\frac{\mathrm{d}}{\mathrm{d}\,\alpha_{\max}} f() = (n-1)\alpha \left(-\frac{1}{{\alpha_{\max}}^2} + \frac{n(n-1)}{((n-1)\alpha_{\max} + \alpha)^2} \right) \tag{5.33}$$

The derivative has a positive root at

$$\alpha = \alpha_{\max} \left(1 - n + \sqrt{n^2 - n} \right) \tag{5.34}$$

Substituting this back into the definition of f yields that f is independent of $\alpha_{\max}$ at that location:

$$f_n := f() = n(3 - 2n) + 2(n-1)\sqrt{n(n-1)} \tag{5.35}$$

Now (f_n) is monotonously decreasing, and

$$\lim_{n\to\infty} f_n = \frac{3}{4}. \tag{5.36}$$

We conclude that [2]

Fact 58. *If users have linear utility functions, the total utility at the Nash equilibrium is at least three quarter of the total utility at the optimal resource allocation.*

5.1.3 Multicast with linear utilities

Let us now generalize this example to a setting that considers multicast. In the simplest model inspired by the one of Feigenbaum et al [26], we assume that there are users sharing a transmission, and that the sharing does not imply any extra cost. To model this, we just have to allow that $x_r = x_{r'}$ for distinct users r, r'. Then the user problem (5.14) turns into

MULTICAST_USER$_r(u_r)$

$$\max \left(\alpha_r \left(\frac{\sum_{r':x_r=x_{r'}} w_{r'}}{w_r + W_r} \right) - w_r \right) \text{ over } w_r \tag{5.37}$$

subject to

$$w_r \geq 0. \tag{5.38}$$

For the derivative, we get

$$\frac{\mathrm{d}}{\mathrm{d}w_r}(u_r() - w_r) = \frac{\alpha_r}{w_r + W_r} \left(1 - \frac{\sum_{r':x_r=x_{r'}} w_{r'}}{w_r + W_r} \right) - 1 \tag{5.39}$$

Now

$$\frac{\mathrm{d}}{\mathrm{d}w_r}(u_r() - w_r) < 0 \tag{5.40}$$

if and only if

$$w_r > S - \frac{S^2}{\alpha_r} - \sum_{r' \neq r : x_{r'} = x_r} w_{r'} \tag{5.41}$$

[2]Added in proof: This is a special case of Theorem 3 of [38] which is scheduled for publication. The proof for linear utility functions is considerably simpler and since given here. For the case of elastic supply, see [37].

(and similarly for equality). Therefore, at the equilibrium, we must have $w_r \geq S - \frac{S^2}{\alpha_r} - \sum_{r' \neq r : x_{r'} = x_r} w_{r'}$ for all r, and equality for all r with $w_r > 0$. However,if r and r' are in one group, equality can hold for both only if $\alpha_r = \alpha_{r'}$. Write $\alpha_r^g = \max_{r' \in G_r} \alpha_{r'}$. Then from (5.14), we get $g_r = S\left(1 - \frac{S}{\alpha_r^g}\right)^+$.

This means that the resource is split between those multicast groups for which α^g is large enough in proportion with the *maximal* utility gradients of each group, that the users with maximal utility gradient in each group pay and the other users enjoy free service on the level that their group leaders are willing to pay.

5.1.3.1 Example with 3 users in 2 groups.

Let us look at a simple example with two groups: g_1 consisting of users 11 and 12 with $\alpha_{11} < \alpha_{12}$, and group g_2 with one user 2 with α_2. With $S = w_{12} + w_2$, we get from (5.20) and claim b of fact 57 for $r = 12$ and $r = 2$

$$w_{12} = \frac{\alpha_{12}^2 \alpha_2}{(\alpha_{12} + \alpha_2)^2} \tag{5.42}$$

$$w_2 = \frac{\alpha_{12} \alpha_2^2}{(\alpha_{12} + \alpha_2)^2} \tag{5.43}$$

and for the utilities

$$u_{11} = \frac{\alpha_{11} \alpha_{12}}{\alpha_{12} + \alpha_2} \tag{5.44}$$

$$u_{12} = \frac{\alpha_{12}^2}{\alpha_{12} + \alpha_2} - w_{12} \tag{5.45}$$

$$= \frac{\alpha_{12}^3}{(\alpha_{12} + \alpha_2)^2} \tag{5.46}$$

$$u_{11} + u_{12} = \frac{\alpha_{12}\left(\alpha_{12}^2 + \alpha_{11}(\alpha_{12} + \alpha_2)\right)}{(\alpha_{12} + \alpha_2)^2} \tag{5.47}$$

5.1.3.1.1 Comparison with group agent. Suppose that the first group from the example above employs a group agent that adjust a weight w_1 used jointly by users 11 and 12. This means that the multicast stream is treated exactly like a unicast stream. We get

$$w_1 = \frac{(\alpha_{11} + \alpha_{12})^2 \alpha_2}{(\alpha_{11} + \alpha_{12} + \alpha_2)^2} \tag{5.48}$$

$$w_2 = \frac{(\alpha_{11} + \alpha_{12}) \alpha_2^2}{\alpha_{11} + \alpha_{12} + \alpha_2} \tag{5.49}$$

$$u_1 = (\alpha_{11} + \alpha_{12}) \frac{w_1}{w_1 + w_2} - w_1 \tag{5.50}$$

$$= -\frac{(\alpha_{11} + \alpha_{12})^3}{(\alpha_{11} + \alpha_{12} + \alpha_2)^2} \tag{5.51}$$

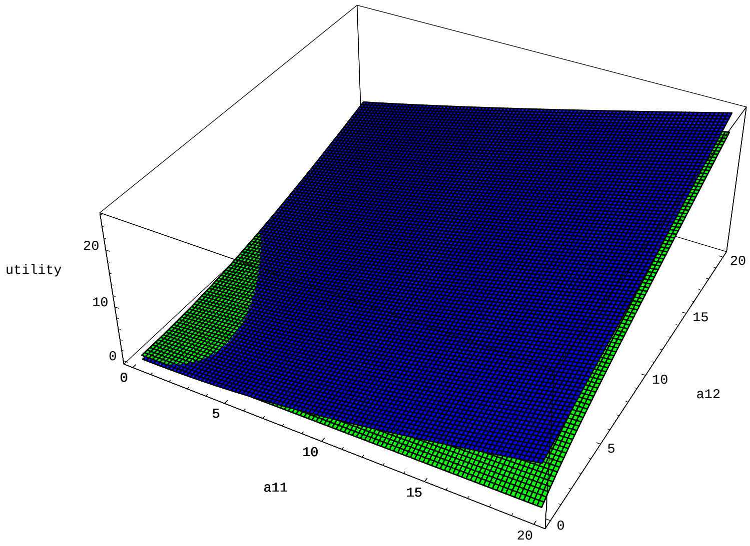


Figure 5.1: Utility without (green) versus with (blue) group agent, a_2=10

From straightforward calculation, we get that the group utility with group agent is larger than group utility without group agent, if and only if

$$(\alpha_{11}, \alpha_{12}, \alpha_2) \in \mathfrak{R}^3_+ \text{ and} \tag{5.52}$$

$$\alpha_2 < \frac{\alpha_{11}^2 + \alpha_{11}\alpha_{12}}{2\alpha_{12}} + \frac{1}{2}\sqrt{\frac{\alpha_{11}^4 + 2\alpha_{11}^3\alpha_{12} + 5\alpha_{11}^2\alpha_{12}^2 + 8\alpha_{11}\alpha_{12}^3 + 4\alpha_{12}^4}{\alpha_{12}^2}} \tag{5.53}$$

$$= \frac{(\alpha_{11} + \alpha_{12})\left(\alpha_{11} + \sqrt{\alpha_{11}^2 + 4\alpha_{12}^2}\right)}{2\alpha_{12}} \tag{5.54}$$

Figure 5.1 compares $u_{11} + u_{12}$ from (5.47) (blue surface) and u_1 from (5.51) (green surface) for the case $a_2 = 10$. Figure 5.2 shows the bounds of the polytope $\{(\alpha_{11}, \alpha_{12}, \alpha_2) \in \mathfrak{R}^3_+ : u_{11} + u_{12} < u_1\}$.

5.1.3.1.2 Remark. One can easily construct an example such that the first group is not served at all even though its total utility is larger than that of both other groups: let the first group consist of three users with $\alpha_{11} = \alpha_{12} = \alpha_{13} = \frac{25}{3}$ (write $\alpha_1 = \alpha_{11} + \alpha_{12} + \alpha_{13}$), let $\alpha_2 = 10$ and introduce a third group with one member with

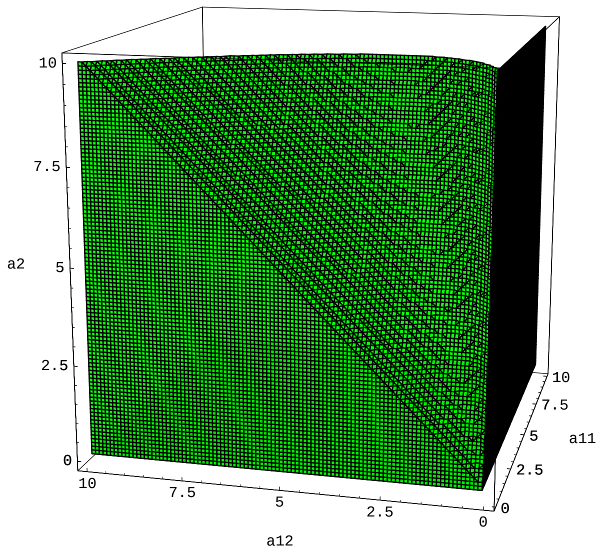


Figure 5.2: Polytope of points where utility with group agent is larger than without

$\alpha_3 = 20$. We get from (5.25) with $a = \frac{\alpha_1\alpha_2\alpha_3}{(\alpha_1\alpha_2+\alpha_1\alpha_3+\alpha_2\alpha_3)^2}$

$$v_{11} + v_{12} + v_{13} = a(\alpha_1\alpha_2 + \alpha_1\alpha_3 - \alpha_2\alpha_3) = -\frac{200}{361} \tag{5.55}$$

$$v_2 = a(\alpha_1\alpha_2 - \alpha_1\alpha_3 + \alpha_2\alpha_3) = \frac{1800}{361} \tag{5.56}$$

$$v_3 = a(-\alpha_1\alpha_2 + \alpha_1\alpha_3 + \alpha_2\alpha_3) = \frac{2200}{361} \tag{5.57}$$

It follows that $w_{11} = w_{12} = w_{13} = 0$ and after another iteration, we compute $w_2 = \frac{400}{81}, w_3 = \frac{500}{81}$.

5.1.3.2 Conclusion.

From the examples above, it follows that allocation at the Nash equilibrium for multicast with individual weighs is arbitrarily inefficient in comparison with the optimal allocation if multicast groups grow large. This contrasts the approximativity of the Nash equilibrium for unicast (fact 58).

5.2 Logarithmic utilities

Suppose now that, as in the original model of Kelly, users have logarithmic utilities:

$$u_r = \alpha_r \log(x_r) - w_r \tag{5.58}$$

If resource allocation is in proportion with the w_r's, then the user problem is

LOGUSER$_r(u_r)$

$$\max\left(\alpha_r\log\left(\frac{w_r}{w_r+W_r}\right)-w_r\right)\text{ over }w_r \tag{5.59}$$

subject to

$$w_r \geq 0 \tag{5.60}$$

Note that $\lim_{w_r\to 0} u_r = -\infty$ and thus at the equilibrium, the derivative of u_r must be zero for every r. In this example, voluntary participation is not satisfied in general in the sense that if a user refuses to pay anything, his resource share will be zero and his utility $-\infty$.

Now with $S=\sum_r w_r$,

$$\frac{\mathrm{d}}{\mathrm{d}w_r}u_r = \frac{\alpha_r}{w_r}\left(1-\frac{w_r}{S}\right)-1 \tag{5.61}$$

and at the equilibrium,

$$w_r = \frac{\alpha_r S}{\alpha_r + S}. \tag{5.62}$$

Summing (5.62) up for all r and dividing by $S \neq 0$, we get

$$1 = \sum_r \frac{\alpha_r}{\alpha_r + S} \tag{5.63}$$

which determines $S > 0$ uniquely.

Note that S is the root of a polynomial of degree $n+1$, where n is the number of users.

5.2.1 Numerical simulation for unicast

Let us assume there are 3 users a, b and c and fix $\alpha_a = 1$ for user a. How does a's optimal weight, resource share and surplus (utility minus costs) vary with b's and c's weight?

Figures 5.3, 5.4 and 5.5 show a's optimal weight , resource share and surplus for b and c varying between 0 and 2.

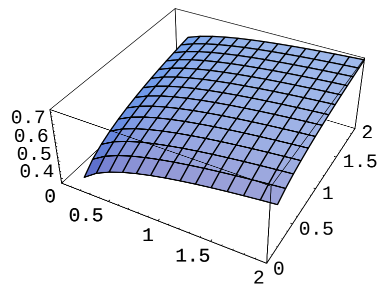


Figure 5.3: Equilibrium weight for user a depending on α_b and α_c

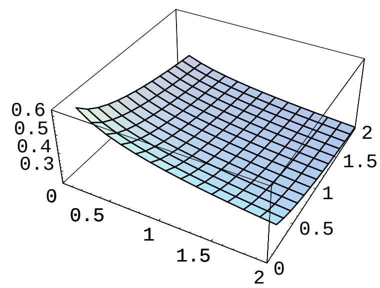


Figure 5.4: Equilibrium resource share for user a depending on α_b and α_c

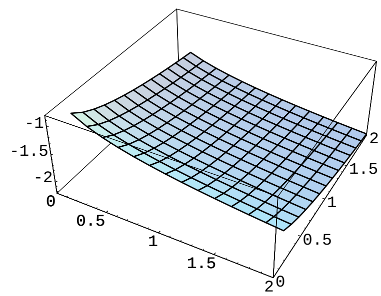


Figure 5.5: Equilibrium surplus for user a depending on α_b and α_c

5.2.2 Approximativity of the Nash equilibrium

Logarithmic utility functions as in (5.58) are unbounded from below. Therefore, theorem 3 of [38] does not apply. Anyway, since user's utility is negative, a *lower* bound of the coordination ratio is of interest.

Let $T = \sum_r \alpha_r$. First note

Fact 59.

$$u_{opt}(\alpha_1, \ldots, \alpha_n) = \sum_{1 \le r \le n} \alpha_r \log \frac{\alpha_r}{T}. \tag{5.64}$$

Proof. Consider the function f_α for $\alpha = (\alpha_1, \ldots, \alpha_{n-1})$ defined by

$$f_\alpha : \{(x_1, \ldots, x_{n-1}) : x_r > 0, \sum_r x_r \le 1\} \mapsto \mathbb{R} \tag{5.65}$$

$$f_\alpha(x_1, \ldots, x_{n-1}) = \sum_{1 \le r < n} \alpha_r \log x_r + \alpha_n \log(1 - \sum_{1 \le r < n} x_r). \tag{5.66}$$

For the partial derivatives of f, we have for $1 \le r < n$

$$\begin{aligned} \frac{\partial f_\alpha}{\partial x_r}(x_1, \ldots, x_{n-1}) &= \frac{\alpha_r}{x_r} - \frac{\alpha_n}{1 - \sum_{1 \le r < n} x_r} \\ &= 0 \end{aligned} \tag{5.67}$$

if

$$\frac{x_r}{1 - \sum_{1 \le r < n} x_r} = \frac{\alpha_r}{\alpha_n}. \tag{5.68}$$

There is exactly one point $\mathbf{x}$ where *all* partial derivatives vanish, namely at

$$\mathbf{x} = (x_1, \ldots, x_{n-1}) \tag{5.69}$$

with

$$x_r = \frac{\alpha_r}{T}. \tag{5.70}$$

Now at f's domain boundary

$$\{(x_1, \ldots, x_{n-1}) : x_r = 0 \text{ for some } r \text{ or } \sum_r x_r = 1\}, \tag{5.71}$$

f has value $-\infty$. It follows that $\mathbf{x}$ is a global maximum.

□

In contract to the case with linear utilities, there is no bound for the coordination ratio depending only on n:

Fact 60. *For any* $B > 0$ *and any* $n \geq 2$*, there are* $\alpha_1, \ldots, \alpha_n$ *such that if for* $r = 1, \ldots, n$*, utility functions are defined by (5.58),*

$$\frac{u_{nash}(\alpha_1, \ldots, \alpha_n)}{u_{opt}(\alpha_1, \ldots, \alpha_n)} > B. \tag{5.72}$$

Proof. Consider first the case $n = 2$. Then (5.63) turns into

$$\frac{\alpha_1}{\alpha_1 + S} + \frac{\alpha_2}{\alpha_2 + S} = 1, \tag{5.73}$$

or

$$S = \sqrt{\alpha_1 \alpha_2}. \tag{5.74}$$

Then

$$w_r = \frac{\alpha_r S}{\alpha_r + S} \tag{5.75}$$

and

$$u_{\text{nash}}(\alpha_1, \alpha_2) = \alpha_1 \log \frac{\alpha_1}{\alpha_1 + S} + \alpha_2 \log \frac{\alpha_2}{\alpha_2 + S} \tag{5.76}$$

and consequently

$$\frac{u_{\text{nash}}}{u_{\text{opt}}} = \frac{\alpha_1 \log \frac{\alpha_1}{\alpha_1 + \sqrt{\alpha_1 \alpha_2}} + \alpha_2 \log \frac{\alpha_2}{\alpha_2 + \sqrt{\alpha_1 \alpha_2}}}{\alpha_1 \log \frac{\alpha_1}{\alpha_1 + \alpha_2} + \alpha_2 \log \frac{\alpha_2}{\alpha_1 + \alpha_2}}, \tag{5.77}$$

and this is unbounded for $\alpha_1 = 1$ and α_2 grows large. This concludes the case $n = 2$.

For the general case, simply add users r for $r > 2$ with $\alpha_r = 0$. The optimal utility does not change by introducing these additional users, and equation (5.62) implies that it doesn't change the Nash utility either. This finishes the proof. □

The coordination ratio can be bounded depending on $L = \frac{\max_r \alpha_r}{T}$:

Fact 61. *With* $L = \frac{\max_r \alpha_r}{T}$*, the following holds:*

$$\frac{u_{nash}(\alpha_1, \ldots, \alpha_n)}{u_{opt}(\alpha_1, \ldots, \alpha_n)} \leq 1 - \frac{1}{L \log L + (1 - L) \log(1 - L)}. \tag{5.78}$$

Proof. Let S be satisfying (5.63). Now

$$\sum_r \frac{\alpha_r}{\alpha_r + T} < 1 \tag{5.79}$$

and consequently

$$S < T. \tag{5.80}$$

It follows

$$
\begin{aligned}
u_{\text{nash}}(\alpha_1, \ldots, \alpha_n) &= \sum_r \alpha_r \log \frac{\alpha_r}{\alpha_r + S} && (5.81)\\
&> \sum_r \alpha_r \log \frac{\alpha_r}{\alpha_r + T} && (5.82)\\
&= \sum_r \alpha_r \log \alpha_r \;-\; \sum_r \alpha_r \log(T + \alpha_r) && (5.83)\\
&> \sum_r \alpha_r \log \alpha_r \;-\; \sum_r \alpha_r \left(\log T + \frac{\alpha_r}{T}\right) && (5.84)
\end{aligned}
$$

using that $\log(x+y) < logx + \frac{y}{x}$ for positive x and y, and thus

$$
u_{\text{nash}}(\alpha_1, \ldots, \alpha_n) > \sum_r \alpha_r \log \frac{\alpha_r}{T} \;-\; \sum_r \frac{{\alpha_r}^2}{T}. \tag{5.85}
$$

The coordination ratio can then be bound by

$$
\begin{aligned}
\frac{u_{\text{nash}}(\alpha_1, \ldots, \alpha_n)}{u_{\text{opt}}(\alpha_1, \ldots, \alpha_n)} &< 1 + \frac{\sum_r {\alpha_r}^2}{T \sum_r \alpha_r \log(\frac{T}{\alpha_r})} && (5.86)\\
&\leq 1 + \frac{T}{\sum_r \alpha_r \log\left(\frac{T}{\alpha_r}\right)} && (5.87)
\end{aligned}
$$

using that $\sum_r {\alpha_r}^2 \leq T^2$ for the last inequality.

Now consider the function f

$$
f(\alpha_1, \ldots, \alpha_{n-1}) = \sum_{1 \leq r < n} \alpha_r \log \alpha_r \;+\; (T - \sum_{1 \leq r < n} \alpha_r) \log(T - \sum_{1 \leq r < n} \alpha_r) \tag{5.88}
$$

defined on the polyhedron P with bounds

$$
\begin{aligned}
\frac{\alpha_r}{T} &\leq L\\
\alpha_1 &\geq \alpha_2\\
\alpha_2 &\geq \alpha_3\\
&\cdots\\
\alpha_{n-2} \geq \alpha_{n-1} &\geq T - \sum_{1 \leq r < n} \alpha_r > 0.
\end{aligned}
$$

We have that for $1 \leq r < n$,

$$
\frac{\partial f}{\partial \alpha_r}(\alpha_1, \ldots, \alpha_{n-1}, T) = \log \alpha_r - \log(T - \sum_{1 \leq r < n} \alpha_r) > 0 \tag{5.89}
$$

for all points in P: f assumes its maximum at the bounds of the polyhedron

$$\begin{aligned}\alpha_1 &= TL\\ \alpha_2 &= T(1-L)\\ \alpha_3 &= \alpha_4 = \ldots = \alpha_{n-1} = 0\\ T - \sum_r \alpha_r &= 0.\end{aligned}$$

It follows that

$$f(\alpha_1, \ldots, \alpha_{n-1}, T) < TL\log(TL) + T(1-L)\log(T(1-L)). \tag{5.90}$$

Continuing from (5.87), we conclude

$$\begin{aligned}\frac{u_{\text{nash}}(\alpha_1, \ldots, \alpha_n)}{u_{\text{opt}}(\alpha_1, \ldots, \alpha_n)} &< 1 + \frac{T}{T\log T \;-\; (TL\log TL + T(1-L)\log(T(1-L)))} &(5.91)\\ &= 1 - \frac{1}{L\log L + (1-L)\log(1-L)}. &(5.92)\end{aligned}$$

□

5.2.3 Multicast with logarithmic utilities

Let us now apply our model of multicast for the case of user utilities being logarithmic. Let us write $G_r = \{r' : x_{r'} = x_r\}$ for the multicast group of user r, and $g_r = \sum_{r' \in G_r} w_{r'}$ for the total weight of that group, and $S = \sum_r w_r$ for the total weight. The user problem then is

MULTICAST_LOGUSER$_r(u_r)$

$$\max\left(\alpha_r \log\left(\frac{g_r}{S}\right) - w_r\right) \text{ over } w_r \tag{5.93}$$

subject to

$$w_r \geq 0. \tag{5.94}$$

For the derivative, we get

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}w_r}(u_r() - w_r) &= \frac{\alpha_r S}{g_r}\left(\frac{1}{S} - \frac{g_r}{S^2}\right) - 1 &(5.95)\\ &= \frac{\alpha_r}{g_r}\left(1 - \frac{g_r}{S}\right) - 1 &(5.96)\end{aligned}$$

Thus the derivative is nonnegative as long as

$$g_r \leq S\frac{\alpha_r}{\alpha_r + S}, \tag{5.97}$$

and zero if equality holds. Similarly as in the case for linear utilities, the latter is the case only for those users r for which α_r is maximal within their group, and consequently, for the remaining users r', we have $w_{r'} = 0$.

Fact 62. *The coordination ratio for multicast users with logarithmic utilities can become arbitrarily bad.*

Proof. Let there be two multicast groups: one with n members and one with only 1 member. Suppose for all users r we have $\alpha_r = 1$. According to fact 59, the optimal utility is

$$\begin{aligned} u_{\text{opt}} &= n \log \frac{n}{n+1} + \log \frac{1}{n+1} && (5.98) \\ &= n(\log n - \log(n+1)) - \log(n+1). && (5.99) \end{aligned}$$

The nash utility is

$$\begin{aligned} u_{\text{nash}} &= n \log \frac{1}{2} + \log \frac{1}{2} && (5.100) \\ &= -(n+1)\log 2. && (5.101) \end{aligned}$$

This implies

$$\begin{aligned} \lim_{n\to\infty} \frac{u_{\text{nash}}}{u_{\text{opt}}} &= \frac{-(n+1)\log 2}{n(\log n - \log(n+1)) - \log(n+1)} && (5.102) \\ &= \infty. \end{aligned}$$

□

5.3 General case for multicast

From the formulation of the multicast user problem for general utility functions (5.14) we get

$$\frac{\mathrm{d}}{\mathrm{d}w_r}\left(u_r\left(\frac{g_r}{w_r + W_r}\right) - w_r\right) = u_r'\left(\frac{g_r}{S}\right)\frac{S - g_r}{S^2} - 1 \tag{5.103}$$

This is positive if and only if

$$u_r'\left(\frac{g_r}{S}\right) > \frac{S^2}{S - g_r} \tag{5.104}$$

The right-hand side of (5.104) is identical for all users in a group. At the equilibrium, only those users r for which $u_r'\left(\frac{g_r}{S}\right)$ is maximal among $r' \in G_r$ have positive weight while the others prefer to benefit from free service.

This means that if multicast users submit individual weights to the mechanism, the resource share they are assigned does not in general increase with the total utility of the multicast group, but rather with the *maximal individual utility*. Thus, the equilibrium solution of the resource share problem is far from efficient.

5.4 Summary

In this chapter, we showed that the classical indirect bandwidth allocation mechanism introduced by Kelly cannot easily be applied to multicast settings without loss of much of its efficiency at equilibrium. If utilities are linear, Kelly's unicast mechanism has coordination ratio of at most $\frac{4}{3}$ while if applied to multicast settings, the coordination ratio is unbounded. In the general case and even in the case of logarithmic utilities, neither the unicast nor the multicast mechanisms have bounded coordination ratio, however, the multicast coordination ratio can't be bound even in terms of the logarithmic unicast coordination ratio.

6 Publish/subscribe systems

6.1 Publish/Subscribe Systems

6.1.1 Introduction

Part of the work presented in this chapter were published in [77] and presented as brief announcement at DISC 2004, and in [75].

In many applications today, conglomerates of independently created components have to be integrated into increasingly complex information systems. It is becoming more and more obvious that for large-scale distributed applications a loosely-coupled event-based style of communication has many advantages: it facilitates the clear separation of communication from computation and eases the integration of autonomous, heterogeneous components into complex systems.

Publish/subscribe systems implement the event-based style: individual processing entities, which we call *clients*, can *publish* information without specifying a particular destination. Similarly, clients express their interest in certain types of information by *subscribing*, so clients can be *producers* and *consumers* at the same time. Information is encapsulated in *notifications* and the *notification service* is responsible for notifying each consumer about all occurrences of notifications which match one of its subscriptions.

6.1.2 Importance for mobile applications

In comparison to classical client/server systems, the publish/subscribe paradigm offers serious advantages in information-driven applications. Here, a client is not obliged to poll a data source for updates – she just subscribes for information she is interested in and gets informed whenever new data is available that fits his subscription. In consequence, a loose coupling is achieved and lots of network traffic can be economized. Applying pub/sub in commercial applications the bandwidth savings can become a substantial argument, especially when clients need to get informed in realtime about updates with bandwidth being expensive and scarce.

Consider for example a wireless network of battery-driven info-nodes with clients connected locally to the nodes by wire. Obviously, the bandwidth between the nodes is restricted and data transmission comes at the cost of valuable battery lifetime. In this scenario, to decide if a client, subscribing to some information, will finally be served, depends on its “utility” from the data. For example, a subscriber employed in a research institute, may have a high utility of news from the scientific community, published by newstickers from different news agencies. Thus, his subscription may get served by the network while another subscription for less important news about sports would not. Obviously, for the network operator in this scenario, finding out over which links a message should be sent is a very expensive task regarding the message and network complexity. Additionally, for encouraging the clients not to lie about their utility, a price for the subscription has to be calculated and charged. This price depends on the costs for the transmission and the utility of other clients served on the same network-path. Of course calculation has to be redone every time some change occurs in the network (e.g. a node (un)subscribes or a publisher (un)advertises).

6.1.3 Why formalization?

There is a considerable amount of work on notification services, and many concrete systems have been designed and implemented (e.g., Siena [14], JEDI [16], etc.). Unfortunately, understanding and comparing these systems is difficult because of differing and informal semantics. Research in the area of publish/subscribe has concentrated on informal analyses and systems offering best-effort functionality. Eugster et al. [22] give an overview about publish/subscribe systems and their relatives. With the increasing popularity of publish/subscribe, however, the need for a formal treatment and for systems guaranteeing more stringent properties is arising. A clear and detailed formalisation allows the behaviour of the system to be described unambiguously and provides a basis for further reasoning, e.g. about the correctness of the system. Formalisations have been proved useful in many areas of distributed computing. However, the specification and verification of distributed systems is a complex task. A variety of techniques (e.g. petri nets, temporal logic, automata) have been proposed, each having its own set of strengths and weaknesses.

Propositional linear temporal logic (PTL) [59, 46] has proved to be a powerful tool to characterise and verify the behaviour of concurrent distributed systems [28]. Fiege, Mühl, and Gärtner [27, 55] introduced a formal specification of publish/subscribe systems using linear temporal logic. In their work, a requirement specification for publish/subscribe systems consisting of safety and liveness properties is introduced. To the authors best knowledge no other formalisation for publish/subscribe systems has been proposed yet. Datta et al. [18] informally state liveness and safety conditions, however, only for static subscriptions and not paying respect to the distributed nature of publish/subscribe systems by implicitly assuming global time. Courtenage [15] offers a description of event types using the λ-calculus, allowing a formal specification of filters. However, system states and correct system behaviour are not addressed in this work.

In this paper, we rewrite the formalism presented in [27, 55] such that it is strictly propositional and extend it to provide message completeness guarantees. In a *message-complete* publish/subscribe system, the system eventually acknowledges every subscription and guarantees the delivery of notifications matching an acknowledged subscription from this time on. In a system without message completeness, the delivery of all notifications matching a subscription is also eventually guaranteed, but the consumer is not aware of the time from which on completeness is guaranteed.

The remainder of this paper is structured as follows: Sect. 6.2 introduces a formal specification for message-complete publish/subscribe systems. Then, we present an implementation framework in Sect. 6.3 that realizes message completeness on top of a system without this guarantee. The approach of separating the development of an axiomatic formalism from the description of a possible implementation enables precise formulation of axioms on the distributed state of publish/subscribe systems, and provable statements on the implementation's properties regarding these axioms.

6.2 Formal specification

6.2.1 Propositional temporal logic and traces

Propositional linear temporal logic (PTL) uses formulas recursively built from atomic propositions, the *elementary state predicates*, which are predicates on the finite set $\mathbb{S}$ of *states*, propositional logical connectors $\vee, \wedge, \neg, \Rightarrow$ and temporal quantifiers $\mathcal{U}, \Box, \Diamond$ and $\bigcirc$.

For our purpose, we are a little more precise about the structure of $\mathbb{S}$: The *state* $s \in \mathbb{S}$ of a system is an assignment $s = s : \mathcal{V} \ni v \mapsto s(v)$ of the state variables $v \in \mathcal{V}$ to some value $s(v) \in \mathit{range}(v)$. Both domain and range of s are assumed to be finite.

The semantics of PTL is defined by the notion of traces. A *trace* σ is a sequence of finitely many states

$$\sigma = s_0^\sigma, s_1^\sigma, \ldots, s_n^\sigma \tag{6.1}$$

An ω-*trace* is an infinite sequence

$$\sigma = s_0^\sigma, s_1^\sigma, \ldots \tag{6.2}$$

of states.

Let Σ be the set of all traces and Σ^* the set of all ω-traces. For $\sigma \in \Sigma, \sigma' \in \Sigma \cup \Sigma^*$, we say that σ' *extends* σ if $\sigma = s_0^{\sigma'}, s_1^{\sigma'}, \ldots, s_n^{\sigma'}$ for some $n \geq 0$. For $\sigma \in \Sigma$, define $\sigma^* = \{\sigma' \in \Sigma^* : \sigma' \text{ extends } \sigma\}$. The collection $\{\sigma^* : \sigma \in \Sigma\}$ of base-open sets induces a topology for the space Σ^*.

Proposition 63. *Let* $\Sigma_0 \subseteq \Sigma$. *Then*

$$\Sigma_0^* = \{\sigma^* \in \Sigma^* : \textit{ if } \sigma \in \Sigma \textit{ such that } \sigma^* \textit{ extends } \sigma\textit{, then } \sigma \in \Sigma_0\} \tag{6.3}$$

is a closed subspace of Σ^**.*

Proof. Let σ be in the closure of Σ_0^*, that is, all initial segments of σ can be extended so that the extension is in Σ_0^*. But then, all initial segments of these extensions are in Σ_0. Therefore all initial segments of σ are in Σ_0. $\square$

Definition 64. *Let* $\Sigma_0 \subseteq \Sigma$ *be a set of traces. We say that* Σ_0 *is*

- closed under initial segments*, if* $s_0^\sigma, \ldots, s_i^\sigma, s_{i+1}^\sigma, \ldots, s_n^\sigma \in \Sigma_0$ *implies that* $s_0^\sigma, \ldots, s_i^\sigma \in \Sigma_0$ *for arbitrary* $0 \leq i \leq n$*,*

- closed under suffixes*, if* $s_0^\sigma, \ldots, s_i^\sigma, s_{i+1}^\sigma, \ldots, s_n^\sigma \in \Sigma_0$ *implies that* $s_i^\sigma, s_{i+1}^\sigma, \ldots \in \Sigma_0$ *for arbitrary* $0 \leq i \leq n$*„*

- closed under stuttering*, if* $s_0^\sigma, \ldots, s_n^\sigma \in \Sigma_0$*, implies that*

$$s_0^\sigma, \ldots, s_i^\sigma, s_i^\sigma, \ldots, s_i^\sigma, s_{i+1}^\sigma, \ldots, s_n^\sigma \in \Sigma_0 \tag{6.4}$$

  for arbitrary $0 \leq i \leq n$*, and*

- closed under skipping states*, if* $s_0^\sigma, s_1^\sigma, \ldots, s_n^\sigma \in \Sigma_0$ *implies that*

$$s_0^\sigma, \ldots, s_{i-1}^\sigma, s_{i+1}^\sigma, \ldots, s_n^\sigma \in \Sigma_0 \tag{6.5}$$

  for arbitrary $0 \leq i \leq n$*.*

By definition, an elementary state predicate applied to a trace $\sigma = s_0^\sigma, s_1^\sigma, \ldots, s_n^\sigma$ always refers to state s_0. For instance, the predicate "$v = v_0$" for some state variable $v \in \mathcal{V}$ and some $v_0 \in$ *range*(v) is true for trace σ iff $s_0(v) = v_0$.

Temporal logic allows us to state properties for a trace by introduction of the additional quantifiers $\mathcal{U}, \Box, \Diamond$, and $\bigcirc$. For some formulas ϕ, ϕ' and $\sigma = s_0^\sigma, s_1^\sigma, \ldots, s_n^\sigma$,

1. $\phi \mathcal{U} \phi'(\sigma)$ holds if either for all i, ϕ holds for the trace $s_i^\sigma, s_{i+1}^\sigma, \ldots$, or there is k such that ϕ' holds for $s_k^\sigma, s_{k+1}^\sigma, \ldots$ and ϕ holds for $s_i^\sigma, s_{i+1}^\sigma, \ldots$ for $i < k$[1],

2. $\Diamond\phi(\sigma)$ holds iff there exists i such that ϕ holds for the trace $s_i^\sigma, s_{i+1}^\sigma, \ldots$,

3. $\Box\phi(\sigma)$ holds iff for all i, ϕ holds for the trace $s_i^\sigma, s_{i+1}^\sigma, \ldots$,

4. $\bigcirc\phi(\sigma)$ holds iff ϕ holds for the trace $s_1^\sigma, s_2^\sigma, \ldots$.

Alpern and Schneider [3, 4] give a definition of safety and liveness conditions in this context, and a topological characterisation of them:

Definition 65. *A predicate* P *is a* safety *predicate if the following holds for* $\sigma \in \Sigma_0^*$*: If for any* $i \geq 0$*, there is* $\sigma' \in \Sigma_0^*$ *such that* σ' *extends* $s_0^\sigma, \ldots, s_i^\sigma$ *and* P *satisfies* σ'*, then* P *satisfies* σ*. A predicate* Q *is a* liveness *predicate if for any* $\sigma \in \Sigma_0$*, there is an extension* $\sigma' \in \Sigma_0^*$ *of* σ *satisfying* Q*.*

[1] Note that our $\mathcal{U}$ is written as $\mathcal{W}$ (*waiting for*) in [46].

Definition 66. *Let Σ_0 be closed under final segments. Q is* absolute liveness *predicate in Σ_0^* if for any ω-trace $\sigma = s_0, s_1, \ldots \in \Sigma_0^*$, if there is an $i \geq 0$ such that Q satisfies $s_i, s_{i+1}, \ldots$, then Q satisfies σ.*

With this definition, it is easy to see that P is a safety predicate exactly if the set of ω-traces satisfying P is closed in Σ^*, and that Q is a liveness predicate, if and only if the set of ω-traces satisfying Q is dense in Σ^*. [73] gives a sufficient syntactical condition of safety predicates:

Theorem 67 (Sistla)**.** *Every elementary state predicate is a safety predicate, and if P and Q are safety predicates, so are $P \wedge Q$, $P \vee Q$, $\circ P$, $\Box P$ and $P \mathcal{U} Q$.*

Sistla gives the following strengthening of safety:

Definition 68 (Sistla)**.** *P is a* strong safety *predicate, if P is a safety predicate and closed under stuttering and skipping states.*

Sistla also defines

Definition 69 (Sistla)**.** *P is an L-*safety *predicate, iff for any $\sigma = s_0^\sigma, s_1^\sigma, \ldots$, P satisfies σ if and only if for all $i \geq 0$, the trace $\sigma' = s_0^\sigma, \ldots, s_i^\sigma, s_i^\sigma, s_i^\sigma, \ldots$ is satisfied by P.*

Proposition 70 (Alpern and Schneider[2])**.** *If P is closed under stuttering, then P is a safety predicate if and only if P is an L-safety predicate.*

6.2.2 Formalising publish/subscribe systems

In [55], a formal specification of publish/subscribe systems has been given by defining axioms about the admissible sequences (traces) of interface operations and client states. We will present a formalism that differs from that one by

a) being strictly propositional, using only predicates on states rather than on state transitions, and

b) giving message completeness guarantee.

6.2.2.1 State variables and Interface

We now define the state variables of message complete publish/subscribe systems. State transitions are triggered by *interface operations op* : $\mathbb{S} \mapsto \mathbb{S}$.

Definition 71. *The* state *of a client c of a publish/subscribe system with message completeness guarantee is determined by the following variables:*

- *the input variable for the publisher's role P_c, the set of notifications n published by c,*

- *the variables for the subscriber's role,*
  - *output* D_c*, the set of notifications that* c *received,*
  - *output* D_c^{dup}*, the set of notifications that* c *received at least twice,*
  - *input* S_c*, the set of active subscriptions of* c*.*
  - *output* S_c^{ack}*, the set of acknowledged subscriptions of* c*.*

The input state *of the system is defined to be the state restricted to the input variables, and similarly, the* output state *is defined.*

Next we define the operations that trigger state transitions. We write the operations as $op : v \mapsto v'$, to be understood as operation op transforming state $v \in \mathbb{S}$ to $v' \in \mathbb{S}$.

Definition 72. *The* interface of a publish/subscribe system with message completeness guarantee *contains the following operations:*

- *operations called from the environment:*
  - *pub*$(c, n) : P_c \mapsto P_c \cup \{n\}$*, client* c *publishes notification* n
  - *sub*$(c, F) : S_c \mapsto S_c \cup \{F\}$*, client* c *subscribes to filter* F
  - *unsub*$(c, F) : S_c \mapsto S_c \setminus \{F\}$*, client* c *unsubscribes from filter* F
- *operations called by the system:*
  - *notify*$(c, n, p) : D_c^{\mathit{dup}} \mapsto D_c^{\mathit{dup}} \cup (D_c \cap \{n\}), D_c \mapsto D_c \cup \{n\},$*, client* c *is notified about* n *coming from publisher* p
  - *ack*$(c, F) : S_c^{\mathit{ack}} \mapsto S^{\mathit{ack}} \cup \{F\}$*, client* c *is notified that from now on, notifications matching* F *will eventually be delivered to* c

The initial state *of the system is defined to be the state* s_{init} *with* $P_c = D_c = D_c^{\mathit{dup}} = S_c = S_c^{\mathit{ack}} = \emptyset$ *for all clients* c*.*

Definition 73. *For a trace* $\sigma = s_0^\sigma, s_1^\sigma, \ldots, s_n^\sigma$*, let the* input-restriction of σ*, denoted by* σ^{input}*, be the sequence of the input states* $\sigma^{\mathit{input}} = s_0^{\mathit{input}}, s_1^{\mathit{input}}, \ldots, s_n^{\mathit{input}}$*. Similarly, we define the notion of* output-restriction*.*

Definition 74. *Let* $\sigma = s_0^\sigma, s_1^\sigma, \ldots, s_n^\sigma$ *be a trace. The* reduction *of* σ *is defined to be the largest subsequence*[2] $\sigma' = s_{k_0}^\sigma, s_{k_1}^\sigma, s_{k_2}^\sigma, \ldots$ *of* σ *such that for all* $i \geq 0$*,* $s_{k_{i+1}}^\sigma \neq s_{k_i}^\sigma$*.*[3] *We say that a trace* $\sigma = s_0^\sigma, s_1^\sigma, \ldots, s_n^\sigma$ *is* input-admissible*, if there is a sequence* $\mathit{op}_0, \mathit{op}_1, \ldots, \mathit{op}_m$ *of interface operations such that the input-restriction of the reduction of* σ *is a subsequence of the input-restriction of* $s_{\mathit{init}}, \mathit{op}_0(s_{\mathit{init}}), \mathit{op}_1(\mathit{op}_0(s_{\mathit{init}})), \ldots$*. Similarly, we define the notion of* output-admissibility*.*

[2]Remember that a sequence $\langle s_i : i \in \mathbb{N}\rangle$ is a *subsequence* of $\langle t_j : j \in \mathbb{N}\rangle$ if there is a strictly monotonous sequence $\langle l_i : i \in \mathbb{N}\rangle$ of natural numbers such that for all i, we have $s_i = t_{l_i}$.

[3]Note that although $\langle k_i : i \in \mathbb{N}\rangle$ is not uniquely determined, σ' is.

Note the following facts:

Fact 75.

- *The operations pub, sub, and ack are idempotent for publish/subscribe systems with message completeness guarantee.*
- $\sigma = s_0^\sigma, s_1^\sigma, \ldots, s_n^\sigma$ *is input-admissible if and only if the sequence of sets* $\langle P_c(s_i^\sigma) : 0 \leq i \leq n\rangle$ *is monotonously increasing for all clients* c.
- $\sigma = s_0^\sigma, s_1^\sigma, \ldots, s_n^\sigma$ *is output-admissible if and only if the sequences of sets* $\langle D_c(s_i^\sigma) : o \leq i \leq n\rangle$ *and* $\langle D_c^{\mathrm{dup}}(s_i^\sigma) : o \leq i \leq n\rangle$ *are monotonously increasing for all clients* c.
- *The set of input admissible traces is closed under initial segments, suffixes, stuttering and skipping states, and so is the set of output admissible traces.*

□

6.2.2.2 Axioms of liveness and safety.

We now present the axioms of message complete liveness and safety.[4]

Definition 76. *We say that a publish/subscribe system satisfies* message complete liveness*, if*

$$\Box[\Box F \in S_Y \Rightarrow \Diamond\Box F \in S_Y^{\mathit{ack}}] \tag{6.6}$$

$$\Box[(\Box F \in S_Y^{\mathit{ack}}) \wedge (n \notin P_X) \Rightarrow (\Diamond(n \in P_X \wedge n \in F) \Rightarrow \Diamond n \in D_Y)] \tag{6.7}$$

Condition (6.6) guarantees that subscriptions which are not subsequently cancelled will eventually be acknowledged. Condition (6.7) says that once a subscription was acknowledged, matching notifications published thereafter will eventually be delivered to the subscriber.

Proposition 77. *Conditions (6.6) and (6.7) are absolute liveness predicates in the sense of definition 66.*

Proof. Let $\sigma = s_0^\sigma, s_1^\sigma, \ldots$ be an ω-trace such that for some suffix $s_i^\sigma, s_{i+1}^\sigma, \ldots$ of σ, condition (6.6) holds. We have to prove that $\Box F \in S_Y \Rightarrow (\Diamond\Box F \in S_Y^{\mathrm{ack}})$ holds for all $s_j^\sigma, s_{j+1}^\sigma, \ldots$. This is clear for $j \geq i$, so let now $j < i$ and suppose that $s_j^\sigma, s_{j+1}^\sigma, \ldots$ satisfies $\Box F \in S_Y$. Then also $s_i^\sigma, s_{i+1}^\sigma, \ldots$ satisfies $\Box F \in S_Y$ and consequently $\Diamond F \in S_Y^{\mathrm{ack}}$, which thus is also satisfied by $s_j^\sigma, s_{j+1}^\sigma, \ldots$. The proof for condition (6.7) is similar. □

[4] Strictly spoken, these are axiom schemata, as they are supposed to hold for any clients X, Y, notifications n, and filter F. Also note that we silently use operations on sets and integers in our formulas. This does not alter expressibility since there are only finitely many states.

Definition 78. *We say that a publish/subscribe system satisfies* message complete safety*, if*

$$\square\big[D_Y^{dup} \quad = \quad \emptyset\big] \tag{6.8}$$

$$\square\Big[n \in D_Y \quad \Rightarrow \quad n \in \bigcup_Y P_Y\Big] \tag{6.9}$$

$$\square\Big[n \notin D_Y \wedge \bigcirc n \in D_Y \quad \Rightarrow \quad \bigcirc n \in \bigcup_Y S_Y\Big] \tag{6.10}$$

Message complete safety means that clients will be only notified about notifications that were published by someone and are matching some subscription of that client, and that there are no duplicate notifications.

As a direct consequence of Sistla's theorem, we have

Proposition 79. *The conditions of definition 78 are safety predicates in the sense of definition 65. They also satisfy strong safety and L-safety.* □

Now we can define message complete correct publish/subscribe systems.

Definition 80. *We say that a publish/subscribe system is* message complete correct*, if for all traces σ of states of the system that are input-admissible, σ is output admissible and satisfies safety and liveness.*

6.3 Implementation

In the last section, we gave an axiomatic description of the desired behaviour of a message complete publish/subscribe system. This section, being titled *implementation*, has to start questioning what implementation of a system does actually mean in this context. Practically, a system is implementable if it can be programmed on some hardware using some programming language. From a theoretical point of view, the implementation of some systems are described via *system specification* as opposed to the *requirement specification* we have given above. There are many techniques usable for system specification. However, one that is closely related with temporal logic is based on *fair transition systems* [46, 47]. Fair transition systems have an intrinsic temporal semantic and therefore can be used to derive requirements (our axioms of liveness and safety) from the system specification.

Moreover, Manna and Pnueli introduce a *simple command-style programming language* (SPL) that, additionally to the standard set of assignment, conditional and loop statements, provides for semaphores and commands for channel-oriented, first-in-first-out asynchronous and synchronous message passing. They give a semantic interpretation of their language by defining an equivalent fair transition system.

6.3.1 Specifying module interfaces

The interaction of a system with its environment is described by *module specifications*. The interface of a module to its environment is defined by the declaration of **in**, **out**, and **external** variables. Additional, a module can declare **local** variables. A module can read and write to **local** variables. Local variables cannot be seen by other modules. **in** variables are read-only for the module, **out** variables write-only. Variables declared **external** can be written to by other modules (if properly declared there).

The *body* of a module contains its system specification, written in SPL. For details, we refer to [46, 47].

So one approach of defining an implementation of a correct publish/subscribe system would be to define a module with the interface specification

module
external in S_c *for every client* c
out S_c^{ack} *for every client* c
external in P_c *for every client* c
out D_c *for every client* c
[*body*]

This listing reflects that the module watches the externally manipulated variables S_c, P_c of subscriptions and notifications and computes from this the states of the subscriptions, that is, S_c^{ack}, and writes out notifications to D_c. Suppose that the body of the module is such that it can be shown that correctness in the sense of definition 80 is always satisfied, is this an implementation of a publish/subscribe system that could be used as a model for a real-life implementation?

We suggest that the answer is no. Indeed, the module specification does not at all reflect the fact that communication between the clients is asynchronous. If subscriptions, notifications and notifications are performed synchronously, the whole implementation gets quite trivially – in particular, there is no need for a requirement specification using temporal logic with its $\Diamond$ modifier at all, since subscriptions and notifications can be propagated immediately.

This is why we explicitly model the network topology of our implementation. So let there be given finite sets of clients $c \in \mathcal{C}$ and notifications $n \in \mathcal{N}$. For every filter $F \subseteq \mathcal{N}$, let there be a constant symbol F.

Nodes in our communication network are called *brokers* $b \in \mathcal{B}$. For every client c, we assume there is a *local broker* $b_c \in \mathcal{B}$ that has bidirectional synchronous communication with c. For a given broker b, let L_b be the set of b's *local client*, that is, the set of clients c for which b is the local broker. Assume that $\mathcal{B}$ is a finite set, and that the set of asynchronous communication channels between the brokers forms an acyclic undirected graph $\mathbb{G}$. Let N_b the set of *neighbour brokers* of b, that is, the set of brokers with whom b has a shared edge in $\mathbb{G}$.

The system specification of our composed module has the form

M=[**module**
// for every client c
external in $\mathit{sub}_{c\rightarrow}$
external in $\mathit{pub}_{c\rightarrow}$
out $\mathit{ack}_{\rightarrow c}$
out $\mathit{notify}_{\rightarrow c}$;
$\|_{b\in\mathcal{B}}\ M_b$
]

where M_b is the module specifying the behaviour of broker b.

We assume that the client state variables of definition 71 behave as given in definition 72.

To reflect the fact that the brokers communicate per asynchronous messaging, we specify two asynchronous channels- one for every direction of message passing- for every edge in $\mathbb{G}$. So for $b' \in N_b$, let $\mathit{channel}_{b\rightarrow b'}$ be the asynchronous communication channel from b to b', and $\mathit{channel}_{b'\rightarrow b}$ the channel from b' to b. Now we can specify the interface of the modules M_b for brokers $b \in \mathcal{B}$:

M_b=[**module**
// for every local client $c \in L_b$
external in $\mathit{sub}_{c\rightarrow b}$
external in $\mathit{pub}_{c\rightarrow b}$
out $\mathit{ack}_{b\rightarrow c}$
out $\mathit{notify}_{b\rightarrow c}$

// for every neighbour broker $b' \in N_b$
external in $\mathit{channel}_{b'\rightarrow b}$ channel
external out $\mathit{channel}_{b\rightarrow b'}$ channel;
[**body**]]

Note that the interface of the module M_b is split into one part (the intersection of M_b's interface with M's) that is visible from outside of the composed module M, and another part that is used only by the other submodules $M_{b'}$.

Our road map is to specify the bodies of the modules M_b such that for the composed module M, the axioms of safety and liveness are modularly valid.

We have not yet defined the types of our variables. We will formally let them have integer type and informally assume a bijections between the "real" ranges of our variables (filters for the *sub* variables, etc) and finite subsets of the integers. This allows us to write in our specification statements like

l_0 : $\mathit{channel}_{b'\rightarrow b} \Rightarrow m$;
l_1: **if** $m = \mathit{sub}(F)$
...

where l_1 stands for $m = i$ for some integer i.

Informally, the variables are of the following types:

$\mathit{sub}_{c\rightarrow b}$ of type *set of filters*
$\mathit{ack}_{b\rightarrow c}$ of type *set of filters*
$\mathit{pub}_{c\rightarrow b}$ of type *notification*

```
notify_{b→c} of type notification
channel_{b→b'} of type channel of messages from alphabet Σ_{b→b'}
```

Let us now define the message alphabets $\Sigma_{b \to b'}$ for neighboured brokers b and b'. While for the communication between clients and their local brokers the interface operations *sub*, *pub*, *unsub*, *ack*, *notify* are used, brokers among themselves use operations that are hidden from the local clients. The following message types are used for inter-broker communication between neighboured brokers b and b':

Definition 81. *The message alphabet $\Sigma_{b \to b'}$ consists of the following messages:*

- *forward(n, p) for notification n and client p,*
- *admin$(\mathcal{S},\mathcal{U})$, and*
- *admin_ack$(\mathcal{S},\mathcal{U})$ where $\mathcal{S}$ and $\mathcal{U}$ are sets of filters.*

Informally, *channel*$_{b \to b'} \Leftarrow$ *forward*(n, p) means that broker b forwards to neighbour broker b' the notification n that was published by client p. *channel*$_{b \to b'} \Leftarrow$ *admin_ack*$(\mathcal{S},\mathcal{U})$ means that broker b requests neighbour broker b' to forward notifications matching some filter in $\mathcal{S}$, but not to forward notifications matching some filter in $\mathcal{U}$. *channel*$_{b \to b'} \Leftarrow$ *admin_ack*$(\mathcal{S},\mathcal{U})$ means that broker b confirms that he is forwarding notifications matching a filter in $\mathcal{F}$ but not those matching a filter in $\mathcal{U}$.

6.3.2 State variables

We will now informally introduce the local state of the brokers. Since (as we will see) the set of possible states is finite, we can add a single local integer variable v to the module specification, assume a 1-1 mapping between states and some integers, and thus are allowed to write statements of the form

```
if (informally described state predicate) then
  change state
endif
```

which can be translated into a statement of the form

```
if (v = v_1 ∨ v = v_2 ∨ ... ∨ v = v_n) then
  v = v'
endif
```

Brokers b maintain

- a private *routing table* T_b containing pairs (F, d) where F is a filter and d is a *destination*, i.e. a local client or a neighbour broker. For a destination d of b, we write $T_b^{|d}$ for $\{F|(F, d) \in T_b\}$.
- a private *pending acknowledgements table* P_b containing an entry for every *admin* message that the broker still needs to acknowledge, and a list of dependent outstanding acknowledgements. Note that for every possible filter

set pair $(\mathcal{S},\mathcal{U})$, there is maximally one entry in P_b, and the number of possible dependencies is finite. Thus, there are only finitely many possibilities for P_b.

- *working copies* of the externally written variables $\mathit{sub}_{c\to b}$ and $\mathit{pub}_{c\to b}$, named $\mathit{sub}_{c\to b}^{\text{work}}$ and $\mathit{pub}_{c\to b}^{\text{work}}$.

Under the assumption that there are only finitely many filters, it is guaranteed that there are only finitely many possible states for every broker.

6.3.3 The Framework Algorithms

Now, we give a description of the bodies of the submodules M_b. We will do that in an informal way, which is justified since there are finitely many modules and every module can assume only finitely many states. Therefore, we can freely use loops and iterators, and can write statements like *send message* m *to all neighbour brokers*.

Fig. 6.1 gives the framework of the body of the module M_b. The state transitions are performed in the subprocedures *processAdminMessage*, *processAdminAckMessage* and *processNotification*. Furthermore, *forward* and *pub* messages are processed by the *processNotification* procedure, *sub*, *unsub* and *admin* messages are processed by *processAdminMessage* procedure, and *admin_ack* messages are processed by *processAdminAck* procedure. These procedures we will describe informally. Translation into the module body is straightforward.

6.3.3.1 The procedure *processNotification*

The procedure *processNotification*(d,n) takes a destination d and a notification n as parameters and does the following:

- It sends a *forward*(n) message to all $\mathit{channel}_{b\to b'}$ variables with $b' \neq d$ for which there is an entry $(F,b') \in T_b$ such that $n \in F$ and
- adds n to all $\mathit{notify}_{b\to c}$ variables for which there is $F \in \mathit{sub}_{c\to b}$ matching n.

6.3.3.2 The procedure *processAdminMessage*

This procedure takes a destination and two filter sets $\mathcal{S}$ and $\mathcal{U}$ as parameters. This procedure is called when *admin* messages from neighbour brokers are received and when subscriptions and unsubscriptions are issued by local clients: If broker b_1 receives *admin*$(\mathcal{S},\mathcal{U})$ from neighbour b_2, *processAdminMessage*$(b_2,\mathcal{S},\mathcal{U})$ is called (line 30). If a new subscription F is issued by a local client c, *administer*$(c,\{F\},\emptyset)$ is called (line 10), and respectively *processAdminMessage*$(c,\emptyset,\{F\})$ for an unsubscription (line 15). *processAdminMessage* calls the *administer* procedure with the same parameters. *administer* encapsulates the applied routing algorithm and triggers changes in the routing configuration. These changes are performed at b_1 in two ways:

1. by transforming his own routing table T_{b_1}, and

2. sending out new *admin* messages to some subset of his neighbours except b_2.

We say (cf.[55]) that the local transformation algorithm satisfies the *restricted change* condition, if $b_3 \neq b_2$ implies that $(T'_{b_1})^{|b_3} = T_{b_1}^{|b_3}$, that is, b_1 does not change his routing behaviour for destinations different from b_2. Furthermore, we say that the algorithm satisfies the *restricted impact* condition if the result of the transformation depends only on the value $T_{b_1}^{|b_2}$.

Let us call $\mathcal{S}$ the *positive part* of the message, and $\mathcal{U}$ the *negative part*. We require the following condition: if *administer*$(d, \mathcal{S}, \mathcal{U})$ is called at broker b with $\mathcal{S} = \emptyset$, then $\bigcup (T'_b)^{|d} \subseteq \bigcup T_b^{|d}$. This condition ensures that only the positive part of an *admin* message can cause $\bigcup T_b^{|d}$ to increase. *administer* returns a pair $(\mathcal{M}_S, \mathcal{M}_U)$ where $\mathcal{M}_S$ and $\mathcal{M}_U$ assign filter sets $\mathcal{M}_S(b_3)$ and $\mathcal{M}_U(b_3)$ to each neighbour broker b_3 of b_1. Broker b_1 sends *admin*$(\mathcal{M}_S(b_3), \mathcal{M}_U(b_3))$ to b_3 if one of these sets are nonempty.

In order to acknowledge subscriptions, brokers will reply to *admin* messages with *admin_ack* messages by the following rule:

- If the incoming *admin* message triggers only *admin* messages with empty positive part, and there are no entries in the pending list, it is immediately replied by *admin_ack*.

- Otherwise, the *admin* messages that are sent out are added to the pending list and dependencies are marked. The reply will be sent as soon as b has received *admin_ack* replies for all marked entries in the pending list.

The case for subscriptions and unsubscriptions of local clients is similar.

There is one point in this algorithm that we have to pay attention to: Suppose b' sends three *admin* messages to b: the first one subscribes b' to F, the second one unsubscribes F, and the third one subscribes F again. Now b sends out *admin* messages accordingly, and waits for acknowledgements in order to acknowledge to b'. But how can b distinguish incoming *admin_ack* messages referring to the first of b''s message, from the ones referring to the third? Note that b can do this distinction, even if there are only finitely many states. Let us explain how this can be done:

Two nodes, front (f) and back (b), are bidirectionally connected via asynchronous communication. Both have the states up *and* down*, and have to obey to the following rules:*

- *never may f be up when b is down,*

- *if f is down, eventually b will be.*

Now f receives synchronous signals go up *and* go down*. Upon receiving* go down*, f must do so immediately and signal acknowledgement. Upon receiving* go up*, f*

must eventually go up and then signal acknowledgement. f can assume only finitely many states.

How to achieve this? f maintains the three Booleans is_hot, last_sent, channel_free. Upon receiving go up *or* go down*, f sets is_hot accordingly to* true *or* false*. In case of* go down*, f changes state and acknowledges. If channel_free, it forwards the signal to b and sets last_sent accordingly. b changes state according to the messages received from b and sends acknowledgement. When f receives acknowledgement for* go up *from b, it checks is_hot and if this is true, assumes state* up *and signals acknowledgement. Otherwise, it sends* go down *to b. If f receives acknowledgement for* go down*, and is_hot is true, it sends* go up *to b. If f has sent no message upon receiving acknowledgement from b, it sets channel_free to true.*

Now b maintains the three flags for every pair $(\mathcal{S},\mathcal{U})$ (of the finitely many pairs that exist), thus guaranteeing that the *admin_ack* it sends to its predecessor b' are never out of date.

6.3.4 Valid routing algorithms

The routing framework described above depends on the implementation of the *administer* procedures and on the initial states for T_b. [55] give classes of *administer* procedures that (with proper initial values for T_b) lead to systems that satisfy the safety and liveness axioms defined there. One can prove that these classes do in fact yield systems which satisfy the stronger axioms given here.

The most trivial example of an implementation of the *administer* procedure is the one that leads to routing by *flooding*: suppose in initial state, brokers forward every notification to all neighbour brokers (except the one that sent it). Suppose the *administer* procedure does nothing, that is, returns $(\emptyset,\emptyset)$ for all inputs. We claim that then, the system is correct in the sense of definition 80. Indeed, it is obvious that our system produces admissible output. Furthermore, since never *admin* messages with nonempty positive part will be triggered, any incoming subscription will be acknowledge at once. Anyway all brokers see all messages, so the second step of *processNotification* ascertains that subscribed notifications will be duly delivered.

Another example of routing is *simple routing*. Here, the *administer* procedures work the following way: *administer*$(b',\mathcal{S},\mathcal{U})$ returns $(\mathcal{M}_S,\mathcal{M}_u)$, where

$$\mathcal{M}_S(b'') = \mathcal{S} \tag{6.11}$$

$$\mathcal{M}_U(b'') = \mathcal{U} \tag{6.12}$$

for all neighbour brokers $b'' \neq b'$ of b. Here, *admin_ack*s will be sent first by the leaves of the network topology, that is by the brokers that have only one neighbour broker. A subscription is answered by an acknowledgement similar to the echo algorithm for message broadcast.

Other implementations of *administer* yield more efficient routing algorithms which avoid flooding notifications or filters into the broker network. Covering-based routing, for example, assumes that it can be detected whether a filter matches a superset of notifications of another filter. The covering test is then used to restrict the forwarding of new and cancelled subscriptions. This effects also acknowledging of subscriptions. In Fig. 6.2, client c_1 had issued subscription F which was already acknowledged by the system. Then, client c_2 issues a subscription G which is covered by F. In this case, b_3 only forwards G to broker b_1 and therefore only waits for b_1 to acknowledge G. Hence, after b_3 received the acknowledgement from b_1 it acknowledges G to b_2 which finally, acknowledges G to c_2.

6.3.4.1 Correctness proofs by decomposition

We claim that with suitable implementation of *administer*, safety and liveness formulas are modularly valid for the composed module M. To prove such a claim for formula ϕ, one has to decompose ϕ into ϕ_b formulas such that for every submodule M_b, the formula ϕ_b is modularly valid for M_b and the validity of all ϕ_b's implies that ϕ is valid. Manna and Pnueli [46, Proposition 4] prove that this is a correct rule for composing modules, and that such a decomposition does always exist for modularly valid formulas.

Let ϕ^{safety} be an instance of one of the three safety conditions from definition 78. For a submodule b, let ϕ_b^{ts} be the formula that expresses the temporal semantics of our system specification. It is quite straightforward to see that the conjunction of the ϕ_b^{ts}'s implies ϕ^{safety}.

The decomposition of liveness properties depends on the implementation of the *administer* procedure. Note that changes in the broker's routing tables T_b depend exclusively on the values that the *administer* procedures return. Mühl gives in [55, Definition 3.4] a sufficient condition for *administer* procedures that lead to correct publish/subscribe systems. The decomposition of the liveness properties are based on Mühl's condition. We omit the details here.

body of M_b
  $T_b = \emptyset$;
  $P_b = \emptyset$;
  $\mathit{sub}^{\mathsf{work}}_{b' \to b} = \emptyset$
  $\mathit{pub}^{\mathsf{work}}_{b' \to b} = \emptyset$
  while (true) **do**
    forall ($c \in L_b$) **do**
      forall ($F \in \mathit{sub}_{c \to b} \setminus \mathit{sub}^{\mathsf{work}}_{c \to b}$) **do**
        // new subscription
        processAdminMessage$(c, \{F\}, \emptyset)$;
        $\mathit{sub}^{\mathsf{work}}_{c \to b} = \mathit{sub}^{\mathsf{work}}_{c \to b} \cup \{F\}$
      endforall
      forall ($F \in \mathit{sub}^{\mathsf{work}}_{c \to b} \setminus \mathit{sub}_{c \to b}$) **do**
        // new unsubscription
        processAdminMessage$(c, \emptyset, \{F\})$;
        $\mathit{sub}^{\mathsf{work}}_{c \to b} = \mathit{sub}^{\mathsf{work}}_{c \to b} \setminus \{F\}$
      endforall
      forall ($n \in \mathit{pub}_{c \to b} \setminus \mathit{pub}^{\mathsf{work}}_{c \to b}$) **do**
        // new notification
        processNotification(c, n);
        $\mathit{pub}^{\mathsf{work}}_{c \to b} = \mathit{pub}^{\mathsf{work}}_{c \to b} \cup \{n\}$
      endforall
    forall ($b' \in N_b$)
      $\mathtt{channel}_{b' \to b} \Rightarrow m$; // non-blocking
      [// begin of atomic block
        if *m is "*$\mathtt{forward}(n)$*" message* **then**
          processNotification(b', n);
        endif
        if *m is "*$\mathtt{admin}(\mathcal{S}, \mathcal{U})$*" message* **then**
          processAdminMessage$(b', \mathcal{S}, \mathcal{U})$;
        endif
        if *m is "*$\mathtt{admin\_ack}(\mathcal{S}, \mathcal{U})$*" message* **then**
          processAdminAckMessage$(b', \mathcal{S}, \mathcal{U})$;
        endif
      ] // end of atomic block
    endforall
  endwhile
endbody

Figure 6.1: Body of module M_b

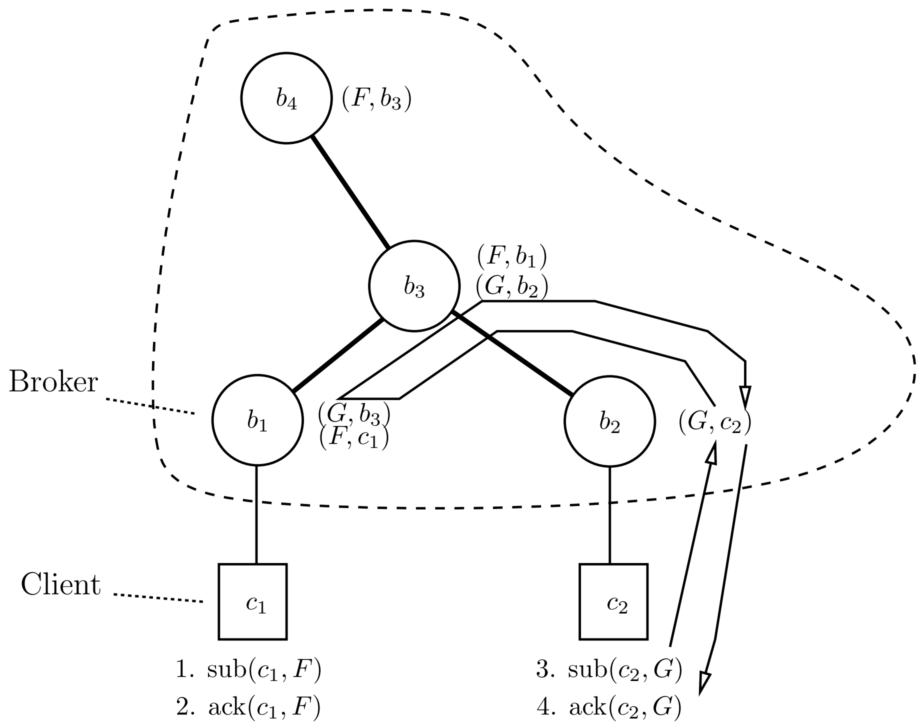


Figure 6.2: Acknowledging a subscription with covering-based routing (F covers G).

6.4 Pricing in publish/subscribe systems

6.4.1 Publish/Subscribe pricing as special multicast pricing

In recent years, lots of research has been conducted about pricing-mechanisms for routing e.g. in ad hoc networks and multicast systems. The algorithms presented in this paper are the first realizing pricing in publish/subscribe systems. We consider scenarios with more than one publisher and a dynamic change at the publisher and subscriber groups. When filters that clients subscribe for overlap, the saved bandwidth is not credited to the clients. We leave this point open for future work. Thus, for simplicity we assume a set of disjoint filters the clients may choose from.

In this scenario, publish/subscribe can be seen as a special type of multicast group communication with two types of groups: *sender groups*, which publish notifications matching certain filters, and *receiver groups*, which are subscribed to selected filters. Clients have "cheap" communication with their local brokers. In particular, we assume that there are no costs for communication between a client and his local broker. On the other hand, communication between brokers is assumed to be "expensive". The costs incurred by inter-broker communication can be modelled in many ways. The simplest approach is to assume a fixed amount per link that is charged whenever the link is used as part of the data transmission. This can be generalized to allow different service levels, or *transmission rates*, with distinct prices for every level. We will go into this point when talking about extensions to multiple rates.

The results in this chapter build on pricing mechanism for multicast groups with one sender and many receivers. Feigenbaum et al. [26] consider marginal cost pricing and Shapley value pricing for mechanisms that compute the optimal receiver subtree of a multicast stream, and the payments of every receiver. Their results were extended by Bläser [7, 8] and Adler and Rubinstein [1] who extended the *all-or-nothing* approach of Feigenbaum et al. and considered the possibility that receivers are served with different service levels, and that there are costs incurred by "enabling" a node to multicast.

Naturally, in the enterprise one has to consider more costs than just the ones caused by the technical infrastructure. For instance, the publishers may charge for the actual contents they transmit to the subscribers. As this paper takes the computer science perspective on business, we do not consider other costs than the ones of data transmission. Also, we assume that the complete broker network is under one administrative domain and payment goes to the network. We do not provide a mechanism how to split the revenue between the network links. On the other hand, every client is seen as an independent agent. It can play both, the role of publisher, and the role of subscriber.

The rest of the paper is structured as follows: Section 6.4.2 introduces the used notation and basic facts. In Section 6.4.3 we present a pricing algorithm for the case where the structure of the network is stable. We extend this algorithm in Section 6.4.4 to cope with changes in the utility of clients and new (un)advertisements. Finally, Section 6.4.6 provides an extensions for multiple rates.

6.4.2 Notation and General Facts

A publish/subscribe systems consists of a set of *clients* which are connected by a network of *brokers*. Every client c is connected to exactly one broker b_c, the *local broker* of c. Let $\mathcal{C}$ be the set of all clients, and $\mathcal{B}$ be the set of all brokers. For a broker b, let L_b be the set of local clients of b. The network is assumed to be undirected and acyclic such that for any two nodes, there is a unique path between them. Communication between clients and their local brokers, and inter-broker communication is based on messages. We assume that message transmission is reliable first-in-first-out.

For communication with the broker network, clients use outbound messages of the types *publish, subscribe, unsubscribe, advertise* and *unadvertise* and inbound *notify* messages. For receiving messages, the client is interested in, she formulates her interest as a filter which she sends in form of a *subscribe* message to its local broker. In our model, a *filter* is some predicate on an attribute of a notification. The filters diffuse through the broker network after a client subscription has been issued based on the routing algorithm. They are used to build up the routing table at each broker. A filter assigns Boolean **true** to a notification if this notifications *matches* the filter (i.e. if the predicate of the filter is true for all considered attributes), or **false** otherwise. *Notifications* are published by clients. A client announces her intention of publishing notifications matching a filter F by sending an *advertise* message, and revokes this announcement with an *unadvertise* message. If a client c subscribes to a filter F, then (after possibly some delay, as formalized by Mühl and Tanner [55, 77]), all notifications that match F, no matter who published them, must be delivered to c via *notify* messages, until the client unsubscribes to this filter.

Let us now fix some filter F. Let $P(F)$ denote the set of clients that have advertised for F. We consider the set $S(F)$ of clients that are subscribed to F as a user group that jointly uses, and shares the costs of, the tree spanned by $P(F)$. We assume that for any link between two brokers, there is a fixed cost of using this link.

Rather than unconditionally subscribing to a filter, our clients send *valuations* for a subscription to some filter F. Client c's valuation for a subscription to F is expressed via its *utility*, a nonnegative real.[5]

Costs, utility and advertisements are defined via *profiles* for a fixed broker topology T and a fixed filter F. A profile for T is a triple $\mathcal{P} = (\xi, u, pub)$, where

- ξ is a function that assigns to every inter-broker link l a nonnegative real $\xi(l)$, the *cost* of l,

[5] We assume a quasilinear setting.

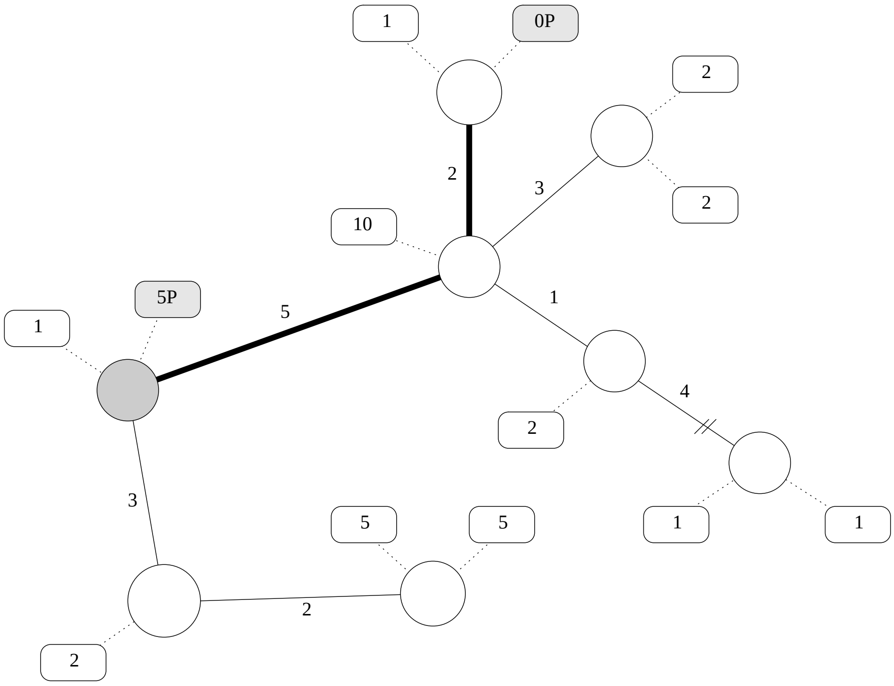


Figure 6.3: Example publish/subscribe network

- u is a function that assigns to every client c a nonnegative real $u(c)$, the *utility* of c from a subscription for F,

- pub is a function that assigns to every client c a Boolean $pub(c)$, true exactly if c has advertised for F.

For a given profile $\mathcal{P} = (\xi, u, pub)$, and a client c, define $\mathcal{P}|_{u(c)=0}$ be the profile that results from $\mathcal{P}$ by changing client c's utility to 0.

Every client c has free connection to its *local broker* b_c. For a set of brokers B, let $T(B)$ be the minimal spanning tree connecting all brokers in B[6], and $\xi(B)$ be cost of that tree, that is,

$$\xi(B) = \sum_{l \in T(B)} \xi(l) \tag{6.13}$$

For a set of clients C, define $T(C) = T(\{b_c : c \in C\})$ the spanning tree connecting all local brokers of clients in C.

Now assume that there is a special *master broker* b^{master}. We view the broker network as a tree rooted at b^{master}. Let $ch(b)$ be the children of b and let $par(b)$ be the parent of b in this tree. Let $Suc(b)$ be the *successor tree* of b, that is, the transitive closure of $\{b\}$ under the ch operation.

[6]Note that since the topology is acyclic, this is well-defined.

Figure 6.3 shows a possible network. Brokers are represented as circles, clients as rounded boxes. The cost of an inter-broker link is printed next to the link. Inside the boxes representing the clients, their utility (for a subscription for a fixed filter) is printed, followed by a "P" if the client has advertised for this filter. The minimal tree connecting all brokers with publishers as local clients is marked by a fat line. The broker presented by the shaded circle functions as master broker.

An *admissible multicast receiver tree* for T is a connected subtree of T that contains $T(\{c : pub(c) = \textbf{true}\})$.

A *mechanism* $\mathfrak{M}$ for T is a function that takes a profile $\mathcal{P}$ as argument, and delivers a pair $(\sigma^{\mathcal{P}}, \pi^{\mathcal{P}})$ where $\sigma^{\mathcal{P}}$ assigns a Boolean[7] $\sigma^{\mathcal{P}}(c)$ to every client c such that $T(\sigma^{\mathcal{P}}) = \{b_c : \sigma^{\mathcal{P}}(c) = \textbf{true}\}$ is an admissible multicast receiver tree, and $\pi^{\mathcal{P}}$ assigns a real $\pi^{\mathcal{P}}(c)$, the *payment* of c, to every client c.[8]

A mechanism $\mathfrak{M}$ satisfies the *no positive transfer* condition, if $\pi(c) \geq 0$ for all clients c and all profiles. $\mathfrak{M}$ satisfies *voluntary participation*, if always $\pi(c) \leq \sigma(c)u(c)$. $\mathfrak{M}$ is *budget-balanced*, if always $\xi(T(\sigma)) \leq \sum_c \pi(c)$. $\mathfrak{M}$ is *strategyproof* if for all $\mathcal{P}$, for all clients c, and for all x

$$\sigma^{\mathcal{P}}(c)u(c) - \pi^{\mathcal{P}}(c) \geq \sigma^{\mathcal{P}^x}(c)u(c) - \pi^{\mathcal{P}^x}(c) \tag{6.14}$$

where $\mathcal{P}^x = \mathcal{P}|_{u(c)=x}$ is the profile used as input by $\mathfrak{M}$ if c pretends to have utility x.

The *social surplus* generated by σ for the profile $\mathcal{P} = (u, \xi, pub)$ is

$$\mathrm{Surplus}(\sigma, \mathcal{P}) = \sum_c \sigma(c)u(c) \; - \xi(T(\sigma)) \tag{6.15}$$

Note that the social surplus has nothing to do with the payment function π of the mechanism.

$\mathfrak{M}$ is a *marginal cost mechanism* (*MC*)[54] if σ maximizes the social surplus, and the payment function π is

$$\pi(c) = u(c) - \left(\mathrm{Surplus}(\sigma, \mathcal{P}) - \mathrm{Surplus}(\sigma, \mathcal{P}|_{u(c)=0})\right). \tag{6.16}$$

It is well-known that this mechanism belongs to the family of Vickrey-Groves-Clarke mechanism and thus is strategyproof. On the other hand, it can easily be seen that it is in general not budget-balanced: if there are at least two local clients connected to every broker and the receiver tree does not change when any one local client changes its utility to zero, then no client will pay anything for transmission.

6.4.3 Marginal Cost Mechanism for Publish/Subscribe Setting: The Static Case

Assume that there is a master broker b^{master} among whose local clients include at least one publisher for F.

[7] Sometimes we will silently cast the Boolean to zero or one by the obvious mapping.

[8] We omit the superscript$^{\mathcal{P}}$ if there is no ambiguity.

Then the algorithm presented in Figures 6.5 and 6.6 shows the marginal cost algorithm for filter F. Note first that the algorithm is a generalization of the one from Theorem 3.1 of [26]: if there is only one publisher, then this publisher plays the role of b^{master} and is the root of the tree, and the flags f^α are not set for all brokers different from b^{master}. It is easy to see that in this case, the algorithm is exactly the one presented by Feigenbaum et al.

Theorem 82. *1. The algorithm shown in Figures 6.5 and 6.6 computes a subtree of the broker network that maximizes the social surplus, that is, the sum of the utilities minus costs, and that is maximal among all trees with this property.*

2. The payments computed in the algorithm are the ones defined by the marginal cost mechanism.

Proof. 1. We prove this by induction on the number of nodes in the tree of all brokers. The claim is trivial if there is only one node. So suppose the claim is proven for trees with less than n nodes. Let T be the computed subtree of a broker tree with less than n nodes, and T^* be an optimal subtree. We have to prove that $Surplus(T) \geq Surplus(T^*)$, and that $T^* \subseteq T$.

Case T is empty, but T^* is not. Since T is empty, we have $\sum_{b^\beta \in ch(b^{\text{master}})} W^\beta < 0$. Now for every b^β, the induction hypothesis implies that $W^\beta \geq Surplus(T^* \cap Suc(b^\beta))$. But then, $\sum_{\beta \in ch(b^{\text{master}})} W^\beta \geq Surplus(T^*)$ and consequently, T wouldn't be empty, a contradiction.

T is nonempty. This is similar: By induction hypothesis, for every child b^β of b^{master}, the computed subtree must be optimal. Therefore, the total surplus of T must be maximal. The maximality of the tree follows from the fact that subtrees with zero utility are included in T.

2. Now we have to prove that the computed payments are the ones from the MC mechanism. Let b^α be some broker in T, and let A be the message b^α sends to its children in Figure 6.6. Then the social surplus of T is decreased by A if we cut from T the subtree rooted at b^α. Now let c be some local client of b^α. If $u(c) \leq A$, then $A - u(c) \geq 0$ and b^α would still be in the tree if c wouldn't participate. Thus c pays nothing in the MC mechanism, and neither does it in our algorithm. Similarly, if $u(c) > A$, then b^α would be cut from the tree (or no multicast would take place at all, if there was a publisher behind b^α), and the marginal costs caused by c are exactly $u(c) - A$.

□

Note that the algorithm requires that the master broker has a publishing client: otherwise, the computed multicast tree may be not optimal. To see this, assume a broker topology as shown in Figure 6.4.3. Here, the optimal tree would contain only b_2 and b_3. The algorithm fails to cut b_1, since the master broker is always served if

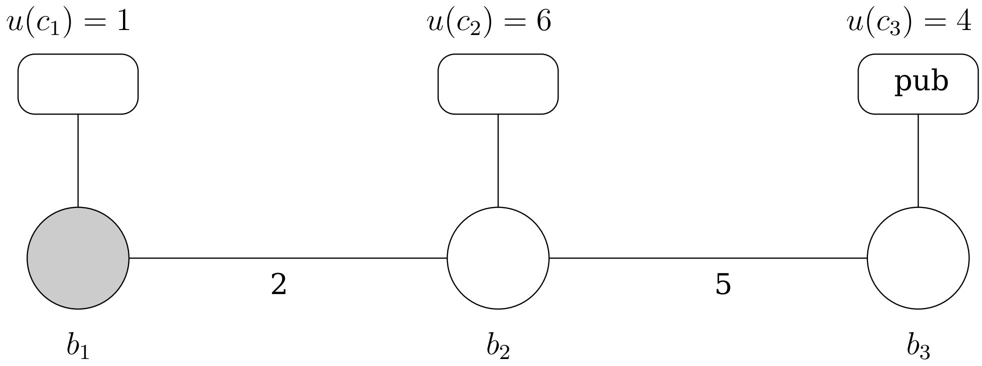


Figure 6.4: Master broker without publishing clients

the receiver tree is nonempty. On the other hand, it is easy to see that if *all* subtrees rooted at the master broker's children contain publishing clients, the computed tree is correct even if the master broker itself doesn't have publishing local clients.

Note also that there are no additional costs for the multi-publisher setting compared to the multicast scenario of Feigenbaum et al.

Corollary 83. *Receiver tree and payments of the marginal cost mechanism can be computed with no more than two messages per link.*

6.4.4 Dynamic Aspects: Changing Utilities and Publishers in the Tree

We are interested in the re-computation of the multicast tree after clients change utility, or advertise or unadvertise a filter. Let us first assume that the change does not concern the property of the master broker being a publisher for the filter. It is easy to see that in the worst case, utility changes or advertisements and unadvertisements have to be propagated through the complete tree. There is, however, an obvious restricted-impact property of the tree:

Fact 84. *Let T be a publish/subscribe network and let $\mathcal{P} = (\xi, u, pub)$ and $\mathcal{P}' = (\xi', u', pub')$ be two cost-utility-publisher profiles for T. Suppose that b^{master} is a broker with $pub(b^{\text{master}}) = pub'(b^{\text{master}}) =$ **true**. Let b^α be a broker such that $T(\{c : pub(c) = true\}) \cap Suc(b^\alpha) = T(\{c : pub'(c) = true\}) \cap Suc(b^\alpha)$. Further suppose that the D sent to the children of b^α is the same for both profiles. Then receiver tree and payments, restricted to $Suc(b^\alpha)$, are the same for both profiles.*

If all publishing local clients of the master broker unadvertise and there is a subtree rooted at one of its children that contains no publishers, a new master has to be found. The algorithm in Figure 6.7 gives a distributed, self-stabilizing computation of the marginal cost tree and payments.

At all brokers b^α

After b^α has received message $A^\beta = (W^\beta, f^\beta)$ from all children $\beta \in ch(b^\alpha)$

/* *The utility of b^α is the sum of the utility of the* */

/* *children* */

l_1: $u^\alpha \Leftarrow \sum_{c \in L_{b^\alpha}} u(c)$

l_2: $W^\alpha \Leftarrow u^\alpha + \sum_{\beta \in ch(b^\alpha), f^\beta = 1 \text{ or } W^\beta > 0} W^\beta - c^\alpha$

/* *Add the node to the receiver set if it has a* */

/* *publishing local client or if one of the* */

/* *descendants has one* */

$f^\alpha \Leftarrow \bigvee_{c \in L_{b^\alpha}} \{c \text{ is publisher for } F\} \vee \bigvee_{\beta \in ch(b^\alpha)} f^\beta$

/* *Send an upward message with the computed* */

/* *values for W^α and f^α to the parent-node* */

$U = (W^\alpha, f^\alpha)$

send U to $par(b^\alpha)$

Figure 6.5: The MC algorithm: Computing the receiver set

Every broker b^α maintains the following state information:

- a variable $par(b^\alpha)$ that contains the current parent node of b^α which is **null** if b^α is master (any neighbour broker different from $par(b^\alpha)$ is a child),
- for every child b^β, the values of W^β and f^β last sent by them,
- the own values of σ^α, W^α and f^α.

A *system state* $\mathfrak{S}$ consists of the collection of the current profile, the states of all brokers, and the set of messages currently on the network. Let us call a system state $\mathfrak{S}$ *correct*, if

- there is exactly one master broker b^{master} whose parent variable $par(b^{\text{master}})$ has value **null**,
- the network, together with the par relation, forms a tree rooted at b^{master},
- the set of brokers b^α with $\sigma^\alpha = 1$ is a maximal surplus-maximizing subtree of T, and
- the prices computed based on W^α are the prices defined by the MC mechanism.

During initialization, an election is performed to choose a master broker. This can be done by an election procedure. Mattern [49] gives (Theorem 2.7 on page 71) an

Initialize
/$*$ *Send D downward the tree with* $*$/
/$*$ $D^{master} = W^{master}$ $*$/
send D to all children
At all brokers b^α
After b^α has received message D from $par(\alpha)$
If ($f^\alpha ==$ **false**)
/$*$ b^α *is not already in the receiver set* $*$/
l_3: $D \Leftarrow \min(D, W^\alpha)$
Endif
If ($D < 0$)
/$*$ *The local clients are not part of the* $*$/
/$*$ *receiver tree* $*$/
send $\sigma = 0$ to all local clients
Else
For every local client c
/$*$ *Calculate the price for every local client* $*$/
l_4: **If** ($u(c) > D$)
$\pi(c) = u(c) - D$
Else
$\pi(c) = 0$
Endif
send $\sigma = 1$ and $\pi(c)$ to c
Endif
For every child $b^\beta \in ch(b^\alpha)$
send D to b^β

Figure 6.6: The MC algorithm: Propagating receiver set and payment

Initialize
Election of new master broker and computation of initial tree
At every broker b_α do forever
/∗ *Has some requirement changed for the* ∗/
/∗ *master-broker?* ∗/
If ($par(b^\alpha)$ == **null** /∗ *b^α is master* ∗/
/∗ *and b^α has no publishing local client* ∗/
and $\{c \in L_{b^\alpha} | pub(c) = \textbf{true}\} == \emptyset$
/∗ *and there is a subtree rooted in a the master's* ∗/
/∗ *broker that has no publishing client* ∗/
and there is $b^\beta \in ch(b^\alpha)$ such that f^β == **false**)
/∗ *Then: Find another master* ∗/
If ($\exists b^\beta \in ch(b^\alpha)$ with f^β == **true**)
/∗ *Choose a child which itself or its subtree has* ∗/
/∗ *a publishing local client* ∗/
l_1: $par(b^\alpha) \Leftarrow b^\beta$
send $U = (W^\alpha, f^\alpha)$ to b^β
Else
/∗ *There is no publisher in the tree anymore* ∗/
go to Initialize
Endif
Endif
/∗ *Has an update message arrived from the child?* ∗/
If (receiving $U = (W^\beta, f^\beta)$ from child $b^\beta \in ch(b^\alpha)$)
/∗ *A state change is reported from a child* ∗/
recompute f^α and W^α
If ($par(b^\alpha)$ == null)
/∗ *The change-information has reached the root* ∗/
send $D = (W^\alpha, f^\alpha)$ to children
Elseif (f^α **or** W^α changed)
send $U = (W^\alpha, f^\alpha)$ to parent
Endif
Endif
/∗ *Has an update message arrived from the parent?* ∗/
If (receiving $U = (W^{\text{par}}, f^{\text{par}})$ from parent $par(b^\alpha)$)
/∗ *A new master is wanted* ∗/
If (b^α has publishing local client)
/∗ *b^α becomes master* ∗/
l_2: $par(b^\alpha) \Leftarrow$ **null**
$W^\alpha \Leftarrow W^\alpha + W^{\text{par}}$
send $D = (W^\alpha, f^\alpha)$ to children
Else
/∗ *look for another master* ∗/
If ($\exists b^\beta \in ch(b^\alpha)$ with f^β == **true**)
l_3: $par(b^\alpha) \Leftarrow b^\beta$
send $U = (W^\alpha, f^\alpha)$ to b^β
Else
/∗ *There is no publisher in the tree anymore* ∗/
go to Initialize
Endif
Endif
Endif

Figure 6.7: Self-stabilizing computation of MC

election algorithm for tree topologies that requires no more than three messages per link. For our purpose, we assume that brokers have a unique, positive ID. Brokers without publishing local clients participate in the election with a zero ID so that it is guaranteed that they won't be elected.

Thereafter, a node changes his parent node in two cases: either the node serves as master but ceases to be eligible, or a node receives a message A from his parent.

We claim that the algorithm has the following property of self-stabilization:

Fact 85. *1. For any publish/subscribe network T and any profile (ξ, u, pub), the algorithm lets converge the system state to a state which is correct in the sense defined above,*

2. After initialization is completed and after a change of the profile, the system state converges again to a correct state.

Proof. Theorem 82 implies the first claim. Now suppose that the system is in a correct state $\mathfrak{S}$, and the profile changes. Let us call a broker b^α a *de-facto-master* if for any neighbour b^β of b^α, we have $par(b^\beta) = b^\alpha$. Then in state $\mathfrak{S}$, the master broker is also de-facto master, and is the only one. Now we claim that there is always exactly one de-facto-master. To see this, note that only at l_1, l_2 and l_3, the de-facto master can change. At l_1 and l_3, the de-facto master just moves to some neighbour. So let us look at l_2. We have to prove that when l_2 is executed, b^α is the unique de-facto master. Now before the execution of l_2, there is a unique de-facto master. Since b^{parent} sent message U to b^α, the de-facto master must be at or behind b^α (seen from b^{parent}). Since $par(b^\alpha) == b^{\text{par}}$ before execution of l_2, the de-facto master must be *at* b^α. Thus, setting $par(b^\alpha)$ to **null** preserves correctness of the state. □

6.4.5 Shapley Value Mechanism

There are two issues about the MC mechanism:

- it is not budget-balanced, and
- although MC is strategyproof, it is not *group-strategyproof*, which means that a group of colluding participants may manipulate pricing to its advantage.

It is well-known [53, 54] for sharing multicast costs with only one sender, that MC is the only strategyproof and efficient mechanism satisfying *consumer sovereignty* (*CS*), *no positive transfers*[9] (*NPT*) and *voluntary participation*[10] (*VP*). Groupwise strategyproof mechanisms satisfying budget-balance and CS, NPT and VP can be characterized as being induced by *cross-monotonic price functions*. Among all

[9] Users will not be paid for receiving a message.

[10] Every user can choose between receiving a message at a cost lower than the utility and not receiving which results in a benefit of 0.

these, the *Shapley value* (*SH*) mechanism is the one that minimizes worst-case welfare loss.

Definition 86. *A* price function *for client* c *is a function* $\pi_c : B \mapsto \pi_c(C) \in \mathbb{R}^+$ *for sets of clients* C*. The price function* π_c *is* cross-monotonic *if* $C \subseteq C'$ *implies* $\pi_c(C) \geq \pi_c(C')$*. Let* $\mathcal{F} = \{\pi_c : c \in \mathcal{C}\}$ *be a family of cross-monotonic price functions. The* mechanism induced by $\mathcal{F}$ *computes*[11] *the receiver set as*

$$C = \lim_{n\to\infty} C_n \tag{6.17}$$

where

$$C_0 = \mathcal{C}\ ;\ C_{n+1} = \{c \in \mathcal{C} : u_c \geq \pi_c(C_n)\}. \tag{6.18}$$

The Shapley value *mechanism is the mechanism induced by the family of price functions* $\{\pi_c : c \in \mathcal{C}\}$ *defined as*

$$\pi_c(C) = \sum_{C' \subseteq C \setminus \{c\}} \left\{ \frac{|C'|!(|C|-|C'|-1)!}{|C|!} \cdot \left[\xi(T(C' \cup \{c\})) - \xi(T(C'))\right] \right\} \tag{6.19}$$

Note that for a fixed filter F,

$$c \in P(F) \wedge C \neq \emptyset \Rightarrow \xi(T(C \cup \{c\})) = \xi(T(C)) \tag{6.20}$$

and thus the cost of the tree spanned by the publishers is shared between all subscribers of F. The costs of the remaining links are shared by all subscribers that are behind it (seen from $T(P(F))$).

Feigenbaum et al. [26] have shown that for a certain class of mechanisms, computing SH requires $\Theta(np)$ messages total and at least p messages over some links, with p clients and n links in the network. This is essentially the complexity of the brute-force mechanism which computes, for each n, the $p_c(S_n)$ in a separate round. They conjecture that this is a lower bound in fact for all computations of SH.

6.4.6 Extension to Multiple Rates

Adler and Rubenstein [1] and Bläser [7, 8] generalize from the all-or-nothing scenario considered so far and allow the multicast transmission to take place using different service levels, or transmission rates. They distinguish between two techniques for providing the transmission rates: the first one uses *layers* built on top of each other. The more layers a transmission uses, the higher rate can be realized. If a transmission uses a certain layer, then it uses also all layers beneath. This implies that two transmissions sharing a link l share the costs of l for all layers that both

[11] Cross-monotonicity implies that the following is well-defined.

of them jointly use, that is, the receiver of the transmission with the higher rate participates in the costs for the lower layers caused by the transmission with the lower rate.

On the contrary, the second technique, called *split paradigm*, defines a new group for every rate. With this technique, transmissions with different rates do not share any resources, even if there is a link used by both of them.

Adler and Rubenstein present their *Max-Layered-Welfare* algorithm that computes, for both layered and split paradigms, the optimal transmission tree. Their algorithm communicates a total number of bits of order $O(\ell h K)$, where ℓ is the number of available layers, h is the height of the network tree and K is the maximal number of bits needed to code a bid. The additional factor h required by Adler and Rubenstein's algorithm is needed because they, in addition to introducing transmission with multiple rates, also introduce costs for *enabling* a node to transmitting an incoming stream to various receivers. Our algorithms 6.5 and 6.6, without supporting layers, require only $O(K)$ bits. We claim that we can modify our algorithm to compute the transmission tree allowing different rates, using $O(\ell K)$ bits.

So let clients submit, instead of a utility $u(c)$, a utility vector $\vec{u}(c)$ where $\vec{u}(c)_j$ is, for $1 \leq j \leq \ell$, c's utility from a subscription to filter F served on level j. Now in Figure 6.5, replace u^α by a vector $\vec{u}^\alpha$ and interpret the sum on the right-hand side of l_1 as vector sum of the $\vec{u}(c)$. In l_2, replace the computation of W^α by

$$\vec{W}_j^\alpha = \max\left\{\vec{W}_{j-1}^\alpha,\ \vec{u}_j^\alpha + \sum_{\beta \in ch(b^\alpha)} \vec{W}_j^\beta - \vec{c}_j^\alpha\right\} \tag{6.21}$$

In Figure 6.6, a broker b^α receives the multicast if the D she sends to his children is nonnegative. We replace D by $\vec{D}$. Broker b^α receives the broadcast on the level j^α that is the largest among those levels l that maximize $\vec{D}_j$. The $\min$ function at line l_3 is to be understood component-wise.

To compute the payment for client c^α, let j^α be the maximum layer c^α is receiving. Write the utility vectors $\vec{u}(c)$ as $\vec{u}^{\text{diff}}(c) = (u_1, u_2 - u_1, \ldots, u_\ell - u_{\ell-1})$ and compute the payments separately for every layer j with $1 \leq j \leq j^\alpha$, as done in Figure 6.6. Add up all these payments. It is quite clear that this is the marginal cost payment for client c^α.

6.4.7 Summary and Outlook

We have demonstrated how to apply marginal cost mechanism to a (specialized) publish/subscribe setting. Algorithms that were known for multicasts with a single source where generalized to a setting with many message sources. A self-stabilizing version of the algorithm was given.

It was shown by Feigenbaum et al. [26] that computing the Shapley value tree is expensive even for multicasts with a single source. It would be nice to know whether admitting multiple sources makes it even harder.

The publish/subscribe setting we considered in this paper is restricted in the sense that we treat subscriptions separately for every filter F. In the case of overlapping filters, the costs for a new subscription may be lower if there are already subscriptions for filters that overlap with the new one. In the current setting, these saved costs are silently swallowed by the network provider. It would be interesting to adopt our mechanisms to cope with overlapping filters.

7 Conclusion and outlook

Game theory is an efficient tool to use when it is necessary to coordinate behaviour of self-interested actors. It finds meaningful applications in network management for autonomous clients. Often the theory will predict that there is an unavoidable loss of efficiency due to the egoistic behaviour of the clients, compared with a network where it can be assumed that (surplus-maximizing) rules will be followed. The aim of this thesis was application, and therefore, the establishment of negative results was not in our focus. Rather, we concentrated on the adoption of known mechanisms to application scenarios in resource management. In the introduction, we stated the thesis that there are four points that make out a good protocol: existence of dominant strategies or equilibria, efficiency at the equilibria, robustness against groupwise strategizing and manageable computational complexity.

We have developed mechanisms of two kinds:

- In chapter 3, we presented pricing schemes and clearing rules for a combinatorial exchange. Continuing the line of research of [58], our pricing scheme resulted from a modification of VGC pricing. Unfortunately, but similarly to the modifications of Parkes et al., the most prominent feature of VGC pricing – truthfulness being a dominant strategy – was lost with that modification. However, a couple of other useful features were preserved, in particular, our pricing rule guarantees that there is never a loss from the acceptance of combinatorial bids. This is a new feature compared to Parkes' modifications. On a broader context, it is an example of a property that – while not directly contributing to global efficiency – is a highly desirable property from the perspective of *some* market participants – in our case, the sellers – which by some reasons have to be honoured because otherwise, they could move to another market that is more profitable for them. Our pricing scheme increases efficiency in comparison to non-combinatorial markets, and it makes combinatorial bids feasible by respecting the seller's interests.

  Also, it was shown that shill bidding always involves a risk of loosing trade. This is one of the few result "against" the possibility of groupwise strategizing.

  The combinatorial exchange setting is the most general exchange setting: arbitrary *interdependencies* like preferred *bundles* can be stated by the users. It is therefore not too surprising that an efficient and budget-balanced mechanism does not exist in that setting.

- The publish/subscribe setting from chapter 6 proves that the situation is much more hopeful if there is a narrower specification of the interdependencies that may occur. In this setting, the *bundles* of goods that users may be interested in are defined by the *filters*. Additionally, there is interdependency generated by cost "savings" by multicast users due to them sharing links. It turns out that in this setting, marginal cost pricing can be applied if budget-balance is not required, and if it is, Shapley value pricing defines a budget-balanced mechanisms that minimizes welfare loss. In this context, the questions of the dominant strategy mechanisms as well as their efficiency therefore is settled.

  While Shapley value mechanisms are even groupwise strategyproof, marginal cost mechanisms are not. We do not know whether there are groupwise strategyproof (of course, *not* budget balanced) mechanisms that are more efficient than Shapley value.

  We also showed that marginal cost prices can be computed with no more than two messages per link (see corollary 83), and we presented a self-stabilizing algorithm that efficiently computes marginal cost prices in dynamic networks (see fact 85).

The issue of strategyproofness has seen a great deal of treatment in the literature and also in this thesis. Much less has been said on the possibility of strategic *groupwise* behaviour. Neither Vickrey pricing, nor marginal cost pricing are robust against groupwise speculation. The negative results on the existence of strategyproof mechanisms immediately show that the situation is hopeless if even groupwise strategizing has to be taken into account. Therefore, the theory is unable to deliver "safe" mechanism. On the other hand, groupwise strategizing requires coordination between the participants, and will often fail due to lack of communication and mutual trust. Participants of a groupwise speculation need a mechanism that splits the benefits they gained between them, and thus encounter the same difficulties as the system they speculate against. It seems that there is a need for *empiric* research on how a mechanism performs if groupwise strategizing is possible. This, however, has completely been left out from this thesis.

While the general theory of mechanism design offers a couple of negative results as well as a quite limited repertoire of standard mechanisms, the design of a mechanism for a real-world application requires analysis of the interplay between resource usage, user's utility optimalization and their limited opportunities of coordination in the situation of the specific application scenario.

Bibliography

[1] Micah Adler and Dan Rubenstein. Pricing multicasting in more practical network models. In *Proc. 13th Ann. ACM-SIAM Symp. on Discrete Algor. (SODA)*, pages 981–990. ACM-SIAM, 2002.

[2] B. Alpern, A.J. Deemers, and F. Schneider. Safety without stuttering. *Information Processing Letters*, 23(4):177–180, 1986.

[3] B. Alpern and F.B. Schneider. Defining liveness. *Information Processing Letters*, 21(4):181–185, 1985.

[4] B. Alpern and F.B. Schneider. Recognizing safety and liveness. *Distributed Computing*, 2(3):117–126, 1987.

[5] A. Archer and E. Tardos. Frugal path mechanisms. In *Proc. 13th Symp. on Discrete Alg*, pages 991–999. ACM/SIAM, 2002.

[6] Aristotle. *Nicomachean Ethics*. Harvard University Press, 1982. ed. by H. Rackham.

[7] Markus Bläser. Budget balanced mechanisms for the multicast pricing problem with rates. In *Proc. 4th ACM Conf. on Electronic Commerce*, pages 194–195, 2002.

[8] Markus Bläser. Budget balanced mechanisms for the multicast pricing problem with rates. Technical report SIIM-TR-A-03-02, Institut für Informatik und Mathematik, Universität Lübeck, 2003.

[9] A. Bonaccorsi, B. Codenotti, N. Dimitri, M. Leoncini, G. Resta, and P. Santi. Realistic combinatorial auctions. In *Proc. IEEE Conference on Electronic Commerce (CEC)*, pages 331–338, Newport Beach, CA, June 2003.

[10] R. Braden, L. Zhang, S. Berson, S. Herzog, and S. Jamin. Resource reservation protocol (rsvp) – version 1 functional specification. RFC 2205 (Proposed Standard), September 1997.

[11] J. I. Bulow and P. D. Klemperer. Auctions vs. negotiations. *American Economic Review*, 86:180–194, 1996.

[12] Lars-Olof Burchard. *Advance Reservations of Bandwidth in Computer Networks*. PhD thesis, Technische Universität Berlin, 2004.

[13] M. Burchardt. *Mikrotheorie. Kritische Einführung mit einem Kompendium mikrotheoretischer Fachbegriffe*. Bund Verlag, Köln, 1986.

[14] Antonio Carzaniga, David S. Rosenblum, and Alexander L. Wolf. Design and evaluation of a wide-area event notification service. *ACM Transactions on Computer Systems*, 19(3):332–383, 2001.

[15] Simon Courtenage. Specifying and detecting composite events in content-based publish/subscribe system. In *22nd International Conference on Distributed Computing Systems Workshops (ICDCSW '02)*, 2002.

[16] G. Cugola, E. Di Nitto, and A. Fuggetta. The JEDI event-based infrastructure and its application to the development of the OPSS WFMS. *IEEE Transactions on Software Engineering*, 27(9):827–850, 2001.

[17] R.M. Cyert and J.G. March. *A behavioral theory of the firm*. Englewood Cliffs, 1963.

[18] A. K. Datta, M. Gradinariu, M. Raynal, and G. Simon. Anonymous publish/subscribe in p2p networks. In *International Parallel and Distributed Processing Symposium (IPDPS'03)*, 2003.

[19] Deutsche Börse Group. *Xetra Stock Market Model*, 2001. http://www.xetra.de.

[20] Deutsche Börse Group. *Xetra Warrant Market Model*, 2001. http://www.xetra.de.

[21] S. DeVries and R. Vohra. Combinatorial auctions: A survey. *INFORMS Journal on Computing*, 15, 2003.

[22] P.Th. Eugster, P. Felber, R. Guerraoui, and A.-M. Kermarrec. The many faces of publish/subscribe. *ACM Computing Surveys*, 35(2):114–131, June 2003.

[23] J. Feigenbaum, C. Papadimitriou, R. Sami, and S. Shenker. A BGP-based mechanism for lowest-cost routing. In *Proceedings of the 2002 ACM Symposium on Principles of Distributed Computing.*, 2002.

[24] J. Feigenbaum and S. Shenker. Distributed algorithmic mechanism design: Recent results and future directions. In *6th International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications*, 2002.

[25] Joan Feigenbaum, Arvind Krishnamurthy, and Rahul Sami. Approximation and collusion in multicast cost sharing. *Games and Economic Behavior*, 47:36–71, 2004.

[26] Joan Feigenbaum, Christos H. Papadimitriou, and Scott Shenker. Sharing the cost of multicast transmissions. *Journal of Computer and System Sciences*, 63(1):21–41, 2001.

[27] Ludger Fiege, Gero Mühl, and Felix C. Gärtner. A modular approach to build structured event-based systems. In *Proceedings of the 2002 ACM Symposium on Applied Computing (SAC'02)*, pages 385–392, Madrid, Spain, 2002. ACM Press.

[28] A. Galton, editor. *Temporal Logics and Their Applications*. Academic Press, 1987.

[29] R. J. Gibbens and F. P. Kelly. Resource pricing and the evolution of congestion control. *Automatica*, 35:1969–1985, 1999.

[30] J.R. Green and J.-J. Laffont. *Incentives in Public Decision Making*. North Holland, 1979.

[31] Albert G. Greenberg, R. Srikant, and Ward Whitt. Resource sharing for book-ahead and instantaneous-request calls. *IEEE/ACM Transactions on Networking*, 7(1):10–22, 1999.

[32] John Hershberger and Subhash Suri. Vickrey prices and shortest paths: What is an edge worth? available at www.cs.ucsb.edu/ suri/psdir/vickrey.ps.

[33] John Hershberger and Subhash Suri. Vickrey prices and shortest paths: What is an edge worth? In *IEEE Symposium on Foundations of Computer Science*, pages 252–259, 2001.

[34] John Hershberger and Subhash Suri. Erratum to "Vickrey pricing and shortest paths: What is an edge worth?". In *IEEE Symposium on Foundations of Computer Science*, page 809, 2002.

[35] John E. Hershberger, Subhash Suri, and Amit M. Bhosle. On the difficulty of some shortest path problems. In *Proc. 20th Symp. Theoretical Aspects of Computer Science (STACS 2003)*, pages 343–354. Springer, 2003.

[36] John Paul II. Centesimus annus. Encyclical, 1991. English version at `http://www.vatican.va/edocs/ENG0214/_INDEX.HTM`.

[37] R. Johari, S. Mannor, and J. N. Tsitsiklis. Efficiency loss in a network resource allocation game: the case of elastic supply. Technical Report 2605, MIT Laboratory for Information and Decision Systems, 2004.

[38] R. Johari and J.N. Tsitsiklis. Efficiency loss in a network resource allocation game. *Mathematics of Operations Research*, 2005. to appear.

[39] S. Kameshwaran and Y. Narahari. A new approach to the design of electronic exchanges. In *EC-Web 2002*, pages 27–36. LNCS 2455, 2002.

[40] F. Kelly. Charging and rate control for elastic traffic. *European Transactions on Telecommunications*, 8:33–37, 1997.

[41] F. Kelly. Fairness and stability of end-to-end congestion control. *European Journal of Control*, 9:159–176, 2003.

[42] F. Kelly, A. Maulloo, and D. Tan. Rate control in communication networks: shadow prices, proportional fairness and stability. *Journal of the Operational Research Society*, 49, 1998.

[43] Paul Klemperer. *Auctions: Theory and Practice*. Princeton University Press, 2004.

[44] Pierre-Simon Laplace. *Essai philosophique sur les probabilités*. Courcier, Paris, 1814.

[45] D. W. Low. Optimal dynamic pricing policies for an m/m/s queue. *Operations Research*, 22, 1974.

[46] Z. Manna and A. Pnueli. *The Temporal Logic of Reactive and Concurrent Systems: Specification*. Springer Verlag, 1992.

[47] Zohar Manna and Amir Pnueli. Temporal specification and verification of reactive modules. Technical report, Weizmann Institute of Science, March 1992.

[48] A. Mas-Colell, W. Whinston, and J. Green. *Microeconomic theory*. Oxford university press, 1995.

[49] Friedemann Mattern. *Verteilte Basisalgorithmen*. Springer, 1989.

[50] Paul Milgrom. *Putting Auction Theory to Work*. Cambridge University Press, 2004.

[51] Bruce L. Miller. Queueing reward system with several customer classes. *Management Science*, 16:234–245, 1969.

[52] Bruce L. Miller and A. G. Buckman. Cost allocation and opportunity costs. *Management Science*, 33(5):626–639, 1987.

[53] H. Moulin. Incremental cost sharing: characterization by strategyproofness. *Social Choice and Wellfare*, 16:279–320, 1999.

[54] H. Moulin and S. Shenker. Strategyproof sharing of submodular costs: budget balance versus efficiency. *Economic Theory*, 18:511–533, 2001.

[55] Gero Mühl. *Large-Scale Content-Based Publish/Subscribe Systems*. PhD thesis, Darmstadt University of Technology, September 2002.

[56] Robert B. Myerson and Mark A. Satterthwaite. Efficient mechanisms for bilateral trading. *Journal of Economic Theory*, 28:265–281, 1983.

[57] Noam Nisan and Amir Ronen. Algorithmic mechanism design. In *Proceedings of 31st Symposium on Theory of Computing*, pages 129–140. ACM, 1999.

[58] David C. Parkes, Jayant Kalagnanam, and Marta Eso. Achieving budget-balance with vickrey-based payment schemes in exchanges. Technical report, IBM Research Report, MAR 2002.

[59] A. Pnueli. The temporal logic of programs. In *Proceedings of the 18th IEEE Symposium on the Foundations of Computer Science*, 1977.

[60] Thomas G. Robertazzi. *Computer Networks and Systems*. Springer, 2nd edition, 1994.

[61] Kevin Roberts. The characterization of implementable choice rules. In Jean-Jacques Laffont, editor, *Aggregation and Revelation of Preferences*, pages 321–348. North-Holland, 1979.

[62] Michael H. Rothkopf, Alexander Pekec, and Ronald M. Harstad. Computationally managable combinatorial auctions. *Management Science*, 44:1131–1147, 1998.

[63] T. Roughgarden and E. Tardos. How bad is selfish routing? *Journal of the ACM*, 49(2):236–259, 2002.

[64] Y Sakurai, M. Yokoo, and S. Matsubara. A limitation of the generalized vickrey auction in electronic commerce. In *Proc. AAAI-99*, pages 86–92, Orlando, FL, 1999.

[65] Paul A. Samuelson. *Economics*. McGraw-Hill, New York, 9th edition, 1973.

[66] T. Sandholm, S. Suri, A. Gilpin, and D. Levine. Winner determination in combinatorial auction generalizations, 2001.

[67] Tuomas Sandholm and Subhash Suri. Improved algorithms for optimal winner determination in combinatorial auctions and generalizations. In *AAAI/IAAI*, pages 90–97, 2000.

[68] O. Schelen and S. Pink. Sharing resources through advance reservation agents. In *Proceedings of the IFIP International Workshop on Quality of Service*, Columbia University, New York, May 1997. Chapman and Hall.

[69] A. Schill, F. Breiter, and S. Kühn. Design and evaluation of an advance reservation protocol on top of rsvp. In *IFIP 4th International Conference on Broadband Communications*, pages 23–40, Stuttgart, 1998. Chapman & Hall.

[70] Jochen Schumann, Ulrich Meyer, and Wolfgang Ströbele. *Grundzüge der mikroökonomischen Theorie*. Springer, 7th edition, 1999.

[71] Oz Shy. *The Economics of Network Industries*. Cambridge University Press, 2001.

[72] Herbert A. Simon. A behavioral model of rational choice. *The Quarterly Journal of Economics*, 69:99–117, 1955.

[73] A. Prasad Sistla. Safety, liveness and fairness in temporal logic. *Formal Aspects in Computing*, 6:495–511, 1994.

[74] Andreas Tanner. On the mean revenue of combinatorial exchanges under variation of clearing policies. In *Proceedings of the 7th International Conference on Business Information Systems BIS*, Poznan, Poland, 2004.

[75] Andreas Tanner and Michael A. Jaeger. Pricing in publish/subscribe systems. In *Proceedings of the 6th International Conference of E-Commerce (ICEC04)*. ACM Press, 2004.

[76] Andreas Tanner and Gero Mühl. A combinatorial exchange for autonomous traders. In *Proceedings of the 4th International Conference on Electronic Commerce and Web Technologies*, volume 2738 of *LNCS*, pages 26–36. Springer Verlag, September 2003. ISBN 3-540-40808-8.

[77] Andreas Tanner and Gero Mühl. A formalisation of message-complete publish/-subscribe systems. Technical Report Rote Reihe 2004/11, Technische Universität Berlin, October 2004.

[78] William Vickrey. Counterspeculations, auctions, and competetive sealed tenders. *Journal of Finance*, 16:8–37, 1961.

[79] Zheng Wang. *Internet QoS: architectures and mechanisms for Quality of Service*. Morgan Kaufmann, 2001.

[80] Lars C. Wolf, Luca Delgrossi, Ralf Steinmetz, Sibylle Schaller, and Hartmut Wittig. Issues of reserving resources in advance. In *Fifth International Workshop on Network and Operating System Support for Digital Audio and Video*, Durham, New Hampshire, USA, 1995.

[81] Elmar Wolfstetter. *Topics in Microeconomics*. Cambridge University Press, 1999.

[82] P. Wurman, W. Walsh, and M. Wellman. Flexible double auctions for electronic commerce: Theory and implementation. *Decision Support Systems*, 24:17–27, 1998.

[83] M. Yokoo, Y. Sakurai, and S. Matsubara. The effect of false name declarations in mechanism design: Towards collective decision making on the internet. In *Proc. 20th International Conference on Distributed Computing Systems (ICDCS-2000)*, pages 146–153, 2000.

[84] Makoto Yokoo, Yuko Sakurai, and Shigeo Matsubara. Robust combinatorial auction protocol against false-name bids. *Artificial Intelligence*, 130:167–181, 2001.

[85] S. Ziya, H. Ayhan, and R. D. Foley. Optimal prices for finite capacity queueing systems. Available at `http://www.unc.edu/~ziya/papers.html`.

[86] Gilead Zlotkin and Jeffrey S. Rosenschein. *Rules of Encounter*. MIT press, 1994.

[87] Edo Zurel and Noam Nisan. An efficient approximate allocation algorithm for combinatorial auctions. In *Proceedings of the 3rd ACM conference on Electronic Commerce*, pages 125–136. ACM Press, 2001.