Abstract

Arguably, the birth of the Internet, the first multi-network communications experiment, took place in November of 1977. This experiment connected the Advanced Research Projects Agency Network (ARPANET) to external sites. Protocols used included TCP (for packet switched routing) and Ethernet (to access a shared resource).

These protocols form the backbone of computing today. E.g., without TCP and without Ethernet, we would have no Internet and no Wi-Fi. One implicit assumption remains unchanged from 1977 to this very day. This is the belief that everyone will follow protocol, and that selfish users will not try to manipulate these protocols.

This assumption makes sense if all users share a common goal, and seek to collaborate with one another towards this goal. While this may be true for the US Defense establishment in 1977, the shared common goal becomes patently absurd when you consider that 32 bit IP address no longer suffice to encompass the number of hosts on the Internet. Furthermore, it is a surprisingly simple matter to manipulate protocols such as TCP and Aloha, for gain, at the expense of others.

In this thesis, we study the use of communications protocols that do not require the good will and altruism of the various network users. In the terminology of Game Theory and Mechanism Design, we study protocols that are incentive compatible. We show just how vulnerable current protocols are to manipulation, to the detriment of all. We suggest alternative protocols that prevent such failures. We also explore realistic models for convergence to equilibria for such communication protocols.

Acknowledgments

First and foremost, I would like to express my gratitude to my supervisors, Prof. Amos Fiat and Prof. Yishay Mansour. Working with these two great men was an honor and a unique intellectual experience. Through numerous meetings Amos and Yishay became both my mentors and my good friends.

I am deeply indebted and thankful to Amos. Amos' trust in me has inspired me with confidence and motivation. I often drew upon his rich experience and interesting insights. His dedication to research, and his open mindedness have been a source of inspiration.

I consider myself very fortunate to have worked with Yishay. Yishay's profound knowledge in algorithmic game theory, as well as in computer science in general was invaluable for my work. His amazing intuition was inspiring, and always helped me focus on the most promising directions.

Amos and Yishay's endless patience and willingness to give advice have immensely contributed to all aspects of my work.

I would like to thank my co-authors Eyal Even-Dar, who contributed to this work, and to Muthu Muthukrishnan, Vahab Mirrokni, and Georgios Piliouras for their part in a later research.

I am thankful to Ilan Gronau, Gera Weiss, Daniel Marco and Edo Liberty for being good friends and advisors. I also want to thank my friends and colleagues at the School of Computer Science.

Last but not least, I would like to thank my family for their love and support. Especially, I thank my wife Shelly for her endless encouragement and love, and for believing in me.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

This thesis addresses the problem of designing protocols for communication networks in which the network users are selfish, and comply with the directives of a protocol only when it is in their best interest to do so. Originally, network protocols were designed under the assumption that users generally follow the protocol "no matter what", a reasonable assumption when networks are small in size, and serve homogeneous users of the same organization. This enabled system designers to focus on optimizing for global objectives, such as the overall utilization of the network, while pushing aside the benefit of individuals.

As networks grew in size, and especially since the rise of the Internet, where numerous agents with different, often conflicting objectives interact, this assumption became problematic. As in many other situations that involve interaction between self motivated agents, it would be more reasonable to assume that each agent is in business for himself; agents optimize their own parameters, be it in the context of maximizing their throughput, minimizing their own delay, or minimizing payments associated with the cost of sending traffic.

Consider for example the Transmission Control Protocol (TCP), the Internet's most prominent and widely used protocol for avoiding congestions. When users transfer data using TCP, they control their transmission rate by setting the size of packets. TCP prescribes slow rate increase at times when there is no congestion, and a swift rate decrease when congestion occurs. Different TCP variants (e.g., [62]) have been proven very effective in coping with congestions. But what if a user is concerned with her own throughput and not with the performance of the entire network? Is following TCP a rational decision from the user perspective?

Perhaps the most appropriate framework for modeling and analyzing a system inhabited by users who act selfishly is within game theory. In this framework, each user is regarded as a *selfish* or a *rational* agent whose actions are intended to maximize her own *utility*. This utility depends on the state of the system, and the system's state is determined by the users' joint action. Informally, a *game* consists of a set of agents, each of which is endowed with a *utility function*, and a set of *strategies*. An outcome of the game is a joint strategy profile consisting of an action for each agent. A *utility* function maps joint strategy profiles to real numbers, with the interpretation that the higher the utility function value is, the better.

For example, in a network setting the utility function of agent Alice could be $\alpha d - w$, where α is a weighting constant, d is the bandwidth rate allocated to Alice and w is the amount of money Alice is required to pay for the allocated bandwidth[1]. In this setting, the action of an agent is the amount of money she is willing to pay, and the allocation is determined by some underlying market mechanism.

Equilibrium analysis plays a major role in game theory and economic theory. The fundamental concept of a *Nash equilibrium* of a game [82] is a joint strategy profile, where no agent can obtain any gain by unilaterally deviating from her current action.

Algorithmic game theory has emerged as a major field of research over the past decade. This new field combines computer science concepts of algorithm design, and complexity with game theory and economic theory. It typically employs the analytic tools of discrete mathematics, and theoretical computer science, such as worst case analysis and approximation ratios, in contrast to classical game theory and economic literature, where it is common to make distributional assumptions regarding the agents. Examples of topics include complexity of equilibrium computation, networking, mechanism design [83], combinatorial auctions [30], online auctions [86], prediction markets, incentives in peer to peer networks, and sponsored search auctions [33]. See [84] for a comprehensive survey on algorithmic game theory.

The inherent heterogeneity and economic interests of Internet users suggests new tools, combining algorithm design, and game theory need be used. Examples of algorithmic game theoretic study of networks include network design [36], routing [91, 12], cost sharing of links [8, 37], resource allocation [63], and bandwidth markets [66, 67].

The Price of Anarchy. A leading approach for analyzing the effect of selfishness is through equilibrium quantification [94], as was first introduced by Koutsoupias and Papadimitriou

[1]See Keshav [68], and Shenker [95] for thorough a discussion on utility functions in a networking environment.

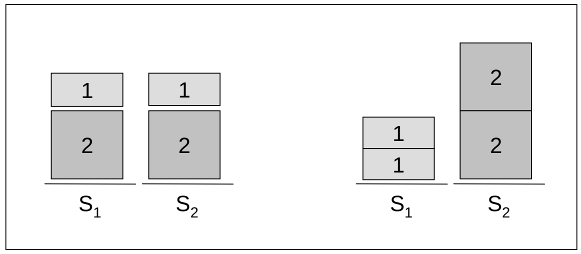


Figure 1.1: A load balancing game with four jobs, two of size 2 and two of size 1, on two machines, S_1, S_2. Both schedules constitute a Nash equilibrium. Notice how no single job can decrease her load by migrating to the other machine. However, the schedule on the left has makespan 3, while the schedule on the right has makespan 4. This already shows that the price of anarchy of this problem is at least 4/3.

[72]. At the heart of this approach lies the assumption that equilibrium is obtained in the long run. The inefficiency in having selfish agents relative to centralized control is then measured by the *price of anarchy*, which is the ratio between the worst possible Nash equilibrium and the social optimum.

$$\text{Price of Anarchy} = \max_{x \in \text{Nash Equilibria}} \frac{\text{Social cost when } x \text{ is played}}{\text{Optimal social cost}}$$

The equilibrium quantification approach had lead to several important results that enhanced understanding of selfish behavior. Koutsoupias and Papadimitriou [72] show lower and upper bounds to the price of anarchy in a selfish load balancing problem. They consider a game where n selfish users have single tasks that are to be scheduled on m machines. Every user seeks to place her task such that the load on the machine chosen is lower than any alternative. The price of anarchy here is defined as the ratio between the makespan[2](social cost) in the worst Nash equilibrium, and the optimal makespan. Figure 1.1 shows Nash equilibria in an instance of this game.

For identical machines, Koutsoupias and Papadimitriou show a tight bound of $2 - \frac{2}{m+1}$ on the price of anarchy, when Nash equilibrium in pure strategies are being considered, and a lower bound of $\Theta(\log m / \log\log m)$ on the price of anarchy when Nash equilibrium in mixed strategies are being considered as well. In a subsequent work, Czumaj and Vocking [31] show a tight bound of $\Theta(\log(m))/\Theta(\log\log(m))$ on the price of anarchy, in a model in which machines are uniformly related, but not necessarily identical.

[2]The makespan of a schedule is defined as the load of the maximally loaded machine.

There is a rich literature on the price of anarchy concerns routing problems [91]. Routing games model the practical problem of routing traffic in a large communication network such as the Internet, where centralized control is unavailable. One such basic model assumes non-atomic agents, where the number of agents is very large (infinite), and each agent controls a negligible fraction of the overall traffic that needs to move from a source to a destination in the network. An agent is assumed to have full control over the path through which she routes her traffic. The cost of routing is the latency accrued along the edges of the route in use, where the latency attributable to an edge depends on the amount of traffic which is routed through it. In their seminal paper, Roughgarden and Tardos [93] show that the price of anarchy in non-atomic selfish routing cannot exceed 4/3, when edges' latency functions are affine; in addition they show that the price of anarchy is independent of the network's size and topology. Subsequently they show that the performance of an optimal flow in any network, does not exceed that of the worst Nash equilibrium in an augmented network with twice the speed (*i.e.*, where the latency of every edge is half its latency in the original network).

Studying the inefficiency of equilibrium in such fundamental networking problems can clarify performance issues when users act selfishly. For example, Roughgarden and Tardos interpret their result as showing that the performance of IP networks (associated with decentralized user control) today, is at least that of ATM networks (associated with centralized authority) a year ago (assuming that network speed doubles every year).

The price of anarchy has been studied in a wide range of other problems. Johari and Tsitsiklis [63] show tight bounds of 4/3 on the price of anarchy in resource allocation problems. Fabrikant et al. [36] study a network creation setting. Anshelevich et al. [8], study games for sharing cost for using network resources.

It is not clear however, how to use these results when designing communication protocols. The price of anarchy measures the performance of a system assuming connections have reached a stable state. Furthermore, the utility of an agent is a function of all the actions of other agents in the system. In order for her to realize that she cannot improve her utility, an agent must be perfectly informed regarding the full state of the network, *i.e.*, every edge cost function, and every route of other agents. In a large scale network such information is unavailable. Furthermore, large networks have a highly dynamic quality, as agents may join or leave the network from, or simply try out different routes from time to time. Studying the price of anarchy of a one shot routing game with full information is more relevant for studying the long term behavior of routing in the network, and probably not sufficiently adequate for the design of network protocols in practice.

1.1 Dynamic Protocols in Equilibrium

In addition to addressing selfish behavior we also want to consider realistic settings, where only limited information is available, and coordination is difficult. This is in contrast to previous work on selfishness in networks.

Coordination: Consider the load balancing game described above. Let Alice and Bob each have a unit size task, and let s_1, s_2 be two identical machines. The assignment A_1, in which Alice assigns her task to machine s_1 and Bob assigns his task to machine s_2 is a Nash equilibrium. Similarly, the assignment, A_2, where Bob's task is scheduled on machine s_1 and Alice's task is scheduled on machine s_2, is a Nash equilibrium as well. In both of the assignments A_1, A_2, neither Alice nor Bob can decrease their cost by switching to the other machine. Moreover, these are the only equilibria in pure strategies. If makespan is used as the system's objective, then both equilibria correspond to an optimal assignment, and the price of anarchy is 1.

In a network setting however, it is likely that Alice and Bob have never met, and cannot agree in advance which of these two equilibria to implement. How does one reach a Nash equilibrium in this case? One possibility is for all the users of the system to agree in advance on a mapping from the set of active users to possible assignments (e.g., when Alice and Bob meet, Alice chooses to s_1, and Bob chooses s_2).

Yet, typically, in a large network, users have little if any information about each other. How can Nash equilibrium be implemented in this case? A possible protocol is for each user to choose uniformly at random one of the machines (this would implement the additional mixed strategy Nash Equilibrium). This protocol is a Nash equilibrium in the sense that if Alice knows that when every user but her follows the protocol, her load is minimized when she following the protocol herself. Is this the most efficient solution?

Lack of Information: Consider a router that seeks to maximize average Quality of Service (QoS), but does not know the QoS requirements of the different channels. This is a case of missing information. Adding payments allows the router to extract the missing information and maximize social welfare.

In this thesis we consider protocols for selfish agents in a variety of networking layers. We design multiple access protocols for establishing selfish usage of physical resources that require exclusive access (in Chapter 3). We consider protocols for routing, and bandwidth allocation (in Chapter 4). Protocols for congestion avoidance are discussed in 5), and we design protocols for regulating traffic in Internet switching points (in Chapter 6).

1.2 Multiple Access Protocols

A multiple-access channel is a broadcast channel shared among multiple users. Users make use of the broadcast channel by sending messages onto the channel. If two or more users simultaneously send messages, then the messages interfere with each other (collide), and are not transmitted successfully. A multiple access channel is not centrally controlled, and to utilize the channel efficiently the users employ a *multiple access protocol*, with the intention of minimizing the chance of collisions (see [88] for a detailed introduction to this subject). Motivating multiple access channels include cabled local area networks (such as Ethernet), and radio local area networks (such as wireless LAN).

Several objectives are relevant when designing a protocol for multiple access. First, a protocol should be *stable*, meaning that the rate at which messages are transmitted successfully, defined as the protocol's throughput, equals the rate at which new messages are generated. Subsequently, the channel capacity is defined as the highest possible value of arrival rate for which, the throughput indeed equals the arrival rate[3]. A second objective is to maximize the throughput. And third, the delay[4] should be minimized.

A typical simplifying assumption in multiple access protocol analysis is slotted time. This means that time is divided into discrete time slots; during any time slot, each user can send at most one message. In acknowledgement-based protocols, the user can determine (by listening to the channel) whether the message was successfully transmitted. In full sensing protocols the user can listen to the channel at every slot, regardless of whether the user transmitted or not, and distinguish between a successful transmission and a "wasted" slot (the binary feedback model) or even more, between a successful transmission, an idle slot, and a collision (the ternary feedback model).

[3]If the arrival rate exceeds the throughput, then the average time it takes a message to leave the system would eventually grow to infinity

[4]The delay of a message is defined as the lapse between a message generation and its successful transmission.

This model has been introduced by Abramson [2], who was confronted with the practical difficulty of ensuring access to the mainframe computers of the University of Hawaii by terminals located in the outer islands of the state. Abramson [2] proposed a simple protocol for resolving collisions, eventually known as the ALOHA protocol. The ALOHA protocol states that a newly generated message is transmitted immediately, hoping for no interference by others. Should the transmission be unsuccessful, every colliding user, independently of the others, re-schedules her transmission to a random time in the future (Randomness is required to ensure that the same messages do not continue to collide indefinitely).

To analyze the performance of Aloha, it is usually assumed that new messages arrive in accordance with a Poisson process, where each message is associated with a new user, that vanishes from the system once her transmission is successful[5]. Under this assumption it can be shown that the capacity of slotted Aloha is $1/e \approx 0.37$. The Poisson assumption is used mainly because it makes the analysis of Aloha-type systems tractable, however, it predicts successfully their maximal throughput. Assuming Poisson arrival process is a basic assumption in the analysis of contention resolution protocols.

The ALOHA protocol has some very attractive properties: it is simple, it does not require the user to listen in slots in which she is idle, and the protocol is "ageless" in the sense that the transmission probability is independent of time. However, it suffers from a significant weakness of being *unstable*, in the sense that overtime, the number of pending messages goes to infinity. More sophisticated age-based protocols exists, where the of transmission probability depends on the some time-counter, and can generally be associate with a sequence $p_1, p_2, \ldots$, of transmission probabilities. A Backoff protocol is also associated with a sequence of transmission probabilities, however, the index t counts the number of times a message collided with others.

Even more advanced is the *full-sensing* family of protocols, where transmission probabilities can be based on the entire channel history, and as a result, be made much more efficient than ALOHA. An important such class of protocols are Collision resolution protocols, in which the users are trying to resolve collisions as soon as they occur. One notable such protocol is the binary tree protocol, described by Tsybakov and Mikhailov [98], and Capetanakis [22].

The binary tree protocol is designed for channels with ternary feedback; it states that

[5]More accurately, for the sake of showing that the capacity of slotted Aloha is $1/e$, it is assumed that the arrival process of new messages and re-scheduled messages is according to Poisson; the two assumptions become close when the re-scheduling is done in a large enough time frame in the future.

when a collision occurs, every user that is not involved in the collision waits for the collision to be resolved, *i.e.*, she seizes transmissions until every user involved in the collision had successfully transmitted. The users involved in the collision split randomly with equal probability into two subsets (by flipping a fair coin). Users in the first subset retransmit in the next slot. The users in the second subset retransmit only after every user in the first subset is successful. Whenever a collision occurs during the resolution of the initial collision, the process repeats recursively.

Basing the transmission probabilities on the personal history of every user, makes breaking the symmetry between users possible, and essentially assigns each user with a unique identity. A version of a binary tree protocols with capacity 0.487 exists, due to Mosely and Humblet [79], and Vvedenskaya and Pinsker [100]. The current best upper bounds, on the capacity that can be achieved by a full-sensing protocolis, is due to Tsybakov and Likhanov [97] who have shown that no protocol can achieve capacity higher than 0.568.

Contention Resolution Protocols for Selfish Agents

In Chapter 3 we concentrate on multiple access protocols for selfish agents. In this respect, our work substantially deviates from classical results in the literature on multiple access protocols. Whereas the utility of a single user may be taken into consideration, in the classical literature (e.g., the delay of an individual user), it is never assumed that the users themselves may deviate from a suggested protocol, in order to maximize their own utility.

Notice that a user entirely controls her transmission probability. It is also clear that both in ALOHA, and in the binary tree protocol, a user that "cheats" during the randomized rescheduling phase, may substantially shorten the delay she experiences. For example, in ALOHA, a user that retransmits immediately after every collision, while the other users follow protocol, would expect a constant delay, rather than a delay which is proportional to the size of the time segment.

We consider a finite set of agents each of which has a single message she must send. We focus on agents with latency cost, that is, the cost of an agent is a non-decreasing function of the time until a successful transmission.

Many challenging problems arise when we assume agents are selfish: (i) At any moment in time, there is an inherent uncertainty from an agent's point of view, regarding the number of other agents that are about to join the game in the future. (ii) Furthermore, even at a specific time slot, the number of currently active agents is kept unknown. (iii) Agents cannot watch directly the actions of other agents; agents act simultaneously, and an agent records only her own action, and the feedback from the channel.

In order to keep the analysis tractable, and to deepen our understanding of the game theoretic aspects of contention resolution, we focus our attention on the resolution of a single collision, and leave the issues of uncertainty regarding the number of active agents, and the uncertainty regarding the arrival of future messages, for future research. We therefore assume that a finite number agents arrive at the same time to the channel.

Channel access is modeled as a game in extensive form with simultaneous play, in which the steps of the game correspond to time slots. At any step of the game, each agent can either *Transmit*, or *be Quiescent*. A mixed strategy is interpreted as the transmission probability in the corresponding time slot. Agents know the channel history (full sensing), and can recall their own actions, but they not know the actual action profile played by the other agents. A strategy is any mapping from channel histories to transmission probabilities, and a protocol is a symmetric profile of behavioral strategies, in equilibrium.

We consider the following natural latency cost functions:

(i) *Deadline costs:* Agents are charged only when they have not been successful prior to a common deadline.

(ii) *Linear latency costs:* The cost of an agent is the same as her delay.

(iii) *Discounted latency costs:* The cost of delay at every step decreases with a fixed discount factor. This method is used to ensure a constant upper bound on the cost of delay.

(iv) *Impatient users:* The cost of delaying an additional time slot increases with the number of time slots already spent in the system. Specifically we consider exponential increase of the cost at every step.

We study equilibria when agents have such cost functions. For the deadline cost function we exhibit an interesting phenomenon of an equilibrium. We show that when the number of available time slots until the deadline is just above some threshold, linear in the number of competing agents, all the agents successfully transmit before the deadline, with high probability. In contrast, when the number of available time-slots until deadline is below some threshold, also linear in the number of agents, then any protocol in equilibrium results

Cost function	Delay of Equilibrium
Linear Time *Independent*	$\Theta(\exp(\sqrt{n})$
Linear Time *Dependent*	$\Theta(n))$, with probability $1 - \exp(-n)$
Discounted	$\exp(n)$
Impatient	$O(n)$; However, a protocol in equilibrium need not exist
Deadline	$O(n)$

Table 1.1: Efficiency of symmetric equilibria in contention resolution games with different cost functions.

without any successful transmission. The threshold phenomenon is clearly observed in Figure 3.3 on page 51, where numerical solutions of the system of equilibrium equations is presented.

For the cost function (ii), (iii), and (iv), we show the existence of a unique protocol in equilibrium when time-independent behavioral strategies are considered. We fully characterize the time-independent equilibrium in the case of the latency cost, and show that transmission probabilities are $O(1/\sqrt{k})$, when k messages are still pending. This implies almost exponentially long delays. For the rest of the cost function we fully characterize the efficiency of equilibrium. Perhaps counter-intuitively, we show that a time-independent equilibrium for agents with discounted latency cost, suffers from exponential delays. On the other hand, we show that the expected delay of impatient bidders is only linear.

We also design a protocol for agents with proportional latency cost, and show it is a Nash equilibrium of the game. Additionally, our protocol is very efficient. With high probability, all n agents successfully transmit within cn time, for some small constant c.

Table 1.1 summarizes our results on contention resolution protocols for selfish agents. A preliminary version of these results appears in [39].

Related Models

Altman et al. [4, 5], study a game theoretic model of slotted Aloha. In their work a very realistic model is studied, where agents have incomplete information as to the number of agents pending. They also assume a stochastic arrival flow to each source. In [5] agents' objective is delay minimization and in [4] agents' objective is to increase their throughput. Agents' strategies are restricted though, to a single retransmission probability. This means that agents do not "learn" from the feedback on the channel, and cannot adjust their behavior accordingly. They show the existence of an equilibrium and give a numerical analysis of the model that shows that the system is inefficient by increasing the delays unduly, even under light traffic.

MacKenzie and Wicker [76] study stability of slotted Aloha, with selfish agents in the multi-packet reception model. They assume that agents utility is a function of the number of attempted transmissions before success, (*e.g.*, costs reflect power lost per successful transmission). They show the existence of equilibrium strategies in this model. They also show that for specific parameters, there exists points of equilibrium that attain the maximum possible throughput of Aloha. Menache and Shimkin [78] also refer to a model where agents are concerned with minimizing their power investment. They provide some lower bounds on the channel capacity that can be obtained in a Nash equilibrium, where agents control their power levels but are assumed to be stationary.

Naor, Raz, and Scalosub [80] study a wireless channel in an interference bound environment, in which, several transmissions may succeed simultaneously, depending on spatial interferences between the different stations. In their model, agents are trying to maximize their success probability at every transmission. They show that when interferences are homogeneous, system performance suffers an exponential degradation in performance at an equilibrium. They also use a penalization scheme for aggressive stations, that ensure that in equilibrium, the system's performance value is a constant fraction of the optimal value.

1.3 Learning to Play Equilibrium

The basic definition of an equilibrium is static: describing a steady state of a system where no agent has an incentive to unilaterally deviate. While an equilibrium can be viewed as a steady state, an important conceptual issue is how such an equilibrium can be reached, and which *dynamics* lead to an equilibrium (see, [49, 102]).

In the simplest model for learning in a game, a set of agents play simultaneously the same game repeatedly. On each step, an agent may learn from past actions and payoffs.

Understanding the dynamics has two different perspectives. The first perspective is developing general dynamics that will always reach an equilibrium (for any game). This line of research includes computationally efficient procedures for correlated equilibrium [44, 56, 24, 17] and inherently exponential time procedures for Nash equilibrium [45, 46, 102, 43, 57, 50] (see, [55] for communication complexity lower bounds). The second perspective is analyzing specific simple and natural behavior and its resulting dynamics in concrete games.

When considering a dynamic behavior, an important and critical question is to what extent is it natural. We are interested in natural dynamics, since they will support the belief that the system naturally reaches an equilibrium. We would like the individual agent procedures to be *rational*, where the agent can be viewed as trying to maximize her long-term utility. We would like the dynamics to be *uncoupled*, where each agent's moves do not depend on the utility functions of the other agents. I.e, every agent is mainly concerned with her own utility. We would like the dynamics to be *simple*, which would support the idea that they naturally occur. Finally, we would like the dynamics to be *flexible* in the sense that different agents may behave differently (and will not have to follow specific prescribed procedure).

One of the most natural and well studied dynamics is *best response*. It prescribes every agent to play a utility maximizing action to the action profile played by the other agents on the previous move. This dynamics dates back to Cournot [29], who showed that in the Cournot oligopoly model with two participating agents, the best response dynamics converges to Nash equilibrium. Rosenthal [90] showed that for every game in the class of congestion games, a sequence of best response moves converges to the Nash equilibrium.

The *fictitious play* dynamics [20] may be regarded as an extension of the best response dynamics, where an agent selects an action which is best response to the average action profile of the other agents. In fictitious play Nash equilibria are absorbing states. That is, if at any time period all agents play a Nash equilibrium, then they will do so for all subsequent rounds. In addition, if fictitious play converges to any distribution, those probabilities correspond to a Nash equilibrium of the underlying game. (see, [49] Proposition 2.2).

In another form of learning dynamics, every agent's objective is to minimize her *regret*. In a regret minimization setting each agent focuses on her own utility (and conceptually is oblivious to the other agents utilities). By *external regret* we compare, in retrospect the agent's average utility to that of the best static action, namely, the best procedure in the class of procedures that use the same action on every step. Having *no-external regret* means that no static procedure would improve significantly the agent's utility.

For example, consider Alice, who drives every morning to work, choosing one of several alternative routes, and recording the time taken. After a year Alice reads in a newspaper report the daily average transit time on every alternative route. Alice's external regret is defined as the difference between her own average transit time, and that of the best fixed route[6].

By *no-internal regret* an agent compares her utility in retrospect using a larger comparison class. Her regret is defined as the difference between her average utility, and the maximum utility she could have had, had she swapped each time she played action a to another action b (where the maximum is taken over all possible pairs of swapping).

The main results in the literature are that there are no-regret algorithm and that their average regret vanishes at the rate of $O(T^{-1/2})$, where T is the number of time steps.

Regret minimization procedures prescribe to most of the requirement we mentioned. It is rational, in the sense that the agent has a guarantee on her own utility regardless of how the other agents act. It is distributed, since an agent needs to be aware only of her own utility. Many of the no-external regret procedures are very simple, and they share the idea that an agent increases the weight on actions that have been doing well. There is a large variety of no-external regret procedures that have been studied, and more conceptually, the assumption is not tied to any specific procedure, but describes the utility of the agents in retrospect.

The regret minimization dynamics have attracted significant attention in recent years. First, the no-internal regret dynamics converges to the set of approximate correlated equilibria [44, 56, 24, 17]. However, for continuous games, no general efficient no-internal regret algorithm is known. Second, there have been works addressing the properties of no-external regret dynamics in specific games. In *z*ero-sum games it is known that the no regret dynamics will converge to the min-max solution [47]. However, for general games, it is known that the no external regret does not necessarily converge to equilibrium [104].

Blum et al. [15] analyze the no-external regret dynamics in routing games showing that most of the agents, most of the time, are playing near equilibrium strategies. Kleinberg et al. [70] study the case where all agents employ a particular no-regret dynamics in a large subclass of atomic congestion games, and show convergence to a pure Nash equilibrium. Blum et al. [16] define the *price of total anarchy* as the ratio of the optimum to that of the worst no-external regret dynamics, and show that it can be similar to the price of anarchy even in cases where the dynamics do not lead to an equilibrium.

[6] This example appears in [32].

Regret Minimization Dynamics in Socially Concave Games

In Chapter 4 we study learning dynamics in a general sub-class of concave games, which we call socially concave games. The class of socially concave games includes many natural, and well studied games, such as (i) Zero-sum games. (ii) Cournot competition [29], a basic economic model for competition among firms. (iii) Resource allocation games [63], that describe bandwidth allocation in networks. (iv) Splittable, atomic routing games [92], where each agent must route her traffic over a congested network, and traffic can be split between different routes.

We focus on regret minimizing dynamics. We show that if each agent follows any external-regret minimization procedure then the dynamics converges in the sense that both the average action vector converges to a Nash equilibrium and that the utility of each agent converges to her utility in that Nash equilibrium.

In addition, for some specific examples of socially concave games we consider the best response dynamics, in which every agent, at every step, plays best response to what the other agents played on the previous step. The best response dynamics is known to diverge for linear Cournot competition, and we show that it also generally diverges for linear resource allocation games, and atomic splittable congestion games.

A preliminary version of our results on socially concave games appears in [35].

1.4 Protocols for Avoiding Congestion

Congestion avoidance algorithms for TCP are the primary basis for congestion control in the Internet, and play an important role in its unprecedented success. The focus of Chapter 5 is a game theoretic analysis of congestion avoidance in data transmission protocols over an IP network.

Congestion is a condition of a severe delay, caused by an overload of packets at one or more switching points (*i.e.*, routers). When congestion occurs, delays increase, and the router begins to enqueue packets until it can send them, and eventually starts to drop packets, when the queue size reaches its storage capacity.

In today's Internet, congestion avoidance over IP networks is done by end-users: a user controls her transmission rate; when a user notices her packets are dropped she adjusts her transmission rate accordingly, in the hope of preventing congestion in the future. In current implementations, a user halves its transmission rate. On the other hand, when her packets are getting through, she increases her rate by one unit. This type of end-to-end congestion control was first proposed by Jacobson [62], and is known as Adaptive Increase, Multiplicative Decrease (AIMD).

The switching point buffering policy also plays some role in congestions avoidance, since users can recognize congestion only by noting packets are being dropped. Thus, a router's dropping policy indirectly affects the rate at which users adjust their transmission rates. Two notable buffering policies are *tail-drop*, and *random early detection*. *Tail-drop* is the simple greedy approach in which packets are dropped only when the router's buffer is full. In *Random Early Detection* [42], packets are dropped before the buffer capacity is reached, in order to signal impending congestion.

The AIMD protocol has been developed in order to promote a social good, namely, efficient and fair use of the network, and is part of today's TCP standard. When the users are selfish the AIMD solution may pose a problem, as it is executed by the users, and basically, if a user wishes to opt out she can do so by modifying her TCP/IP code[7]. When users are selfish, it is natural to ask whether there exists a protocol, hopefully efficient, in equilibrium, which users follow because it is in their best interest to do so, and not out of altruism.

Kelly [66] models the congestion avoidance problem as a flow problem over a capacitated network, where agents buy bandwidth along network routes, and the network sets the prices for a bandwidth share at every edge. Kelly shows that a price equilibrium exists in this framework, and furthermore, this equilibrium attains several desired properties, such as fairness and efficiency. Kelly et al. [67] provide a heuristic argument that AIMD quickly stabilizes send rates to levels corresponding to a fair and efficient sharing of network resources. Low and Lapsley [75] show that when users employ Jacobson's protocol [62], and switches are maintained using the Random Early discard policy [42], then in the limit, traffic flows converge to an optimal solution of Kellys flow market.

Karp et al. [65], attempt to explain the individual rationale for using congestion avoidance protocols, through the study of selfish responses to congestion control. They propose a simple framework, where adversary to model the unpredictable bandwidth availability from the viewpoint of a single agent. In their model, an agent makes sequential decisions on her transmission rate, with lack of information regarding the available capacity at every step,

[7] Modifying ones TCP code is not a trivial task. However, in some operating systems, e.g., Windows and Unix, changing some parameters of TCP, such as the TCP timeout, can be done easily.

based entirely on a binary feedback that tells him in the aftermath, whether the channel has been congested or not.

On the one hand, an agent wishes to use the available capacity to its full, but on the other hand, using more than the available capacity results in some degraded service. Karp et al. model this through a utility function that, at every step, yields a payoff which is a function of her transmission rate, and the available capacity. In their model the available capacity is adversarial, and competitive ratio is used as a performance measure for several agent policies.

Arora and Brinkman [10] present an almost optimal policy for controlling the rate of a single agent using a Multiplicative Increase, Multiplicative Decrease approach. Kesselman and Mansour extend the original model in [65] for multiple users, and present an almost optimal policy, for maximizing social good, which is based on an adaptive version of AIMD.

Regret Based Congestion Avoidance Protocols

In Chapter 5 we propose a game theoretic study of congestion avoidance protocols. We suggest a simple game, that follows the work of Karp et al. [65], to model the interaction between users that transfer data through a common bottleneck. We assume that users' utility is linear in their good-put, *i.e.*, the amount of useful data they manage to transfer, with a penalty, linear in the amount of transfer that exceeds the available capacity; this corresponds to the *gentle utility function* studied in Karp et al. [65]. The main difference between our model and Karp et al. [65] is that in our model the available capacity of an agent is determined by the actions of other agents, and not by an adversary. The strategic approach had been suggested in Karp et al. [65], however, they do not consider any specific rule for allocating bandwidth to agents.

We study the congestion avoidance game with several router's policies for packets dropping, that is, a rule for mapping between a vector of the agents' transmission rates, and an allocation of rates to agents. The policies considered in our work correspond to known buffer overflow policies such as (i) tail-drop, (ii) random early detection, and (iii) fair queueing.

We show that a unique Nash equilibrium exists in this class of congestion avoidance games. We use tools from Chapter 4 to study protocols with a no-regret property. We focus our attention on *projected gradient descent* protocols, for which the no-regret property holds (see, [103]). We show that in a congestion avoidance game, when all agents adjust their transmission rate in accordance with some *projected gradient descent* protocol, their average joint play converges to a Nash equilibrium.

A preliminary version of our results on congestion avoidance for selfish agents appears in [35].

1.5 Queue Management

One of the main bottlenecks in the traffic flow inside communication networks is the management of queues in the connection points, such as switches and routers. If incoming traffic from several sources is directed toward the same destination, it may be impossible to immediately direct all the traffic toward the outgoing link, since the bandwidth of the outgoing link is limited. Packet loss is therefore unavoidable.

The traffic in communication networks tends to arrive in bursts, which is the motivation for buffering the packets in queues, located either at the incoming links, or the outgoing links, or both. The packets arriving in a burst are stored in queues, and are later sent in a speed determined by the bandwidth of both outgoing links and the backbone of the connection device. If traffic is not too heavy, the queues will drain before more bursts arrive, avoiding packet loss. This best effort approach relies on statistical characteristics of communication traffic, assuming that the links are sometimes idle and not always used to their full capacity.

Several types of communications, and especially real time application such as voice and video, cannot rely on best effort, as they are highly sensitive to delays and jitter, and would usually require Quality of Service (QoS) guarantees on bandwidth and delays.

QoS could be thought of as a contract between a service provider and a user. There are two main approaches for attaining QoS on the Internet. The first is *premium service* [14], in which the traffic is shaped upon the entry to the network. Contracts with users are formed so that the service provider could always fulfil her agreements. Packets that are not part of the premium service are given best-effort service. Thus, it is provisioned according to peak capacity profiles. The second approach is *assured service* [27], which is provisioned according to expected capacity usage. Assured service relies on statistical multiplexing that allows for overbooking of service, based on the assumption that usually the users do not use the same resource all at once. In the worst case that the demand for the resource overflows, assured service specifies a priority based guarantees to users.

Differentiated services [27] is an approach for a implementing the spirit of assured service in IP networks. This approach is based on a simple, coarse grained mechanism for classifying packets. The service provider treats packets from different classes differently. Differentiated services, for example, can be used to provide low-latency, guaranteed service to critical network traffic such as voice or video while providing simple best-effort traffic guarantees to non-critical services such as web traffic or file transfers. Additionally, a service provider may decide to treat differently packets of different paying customers (in a pay more - get more fashion).

We consider a model with a single queue. Time is slotted, and a sequence of packets arrive at the queue, each of which has an associated QoS class it belongs to. A queue manager admits packets online. Whenever a packet is sent, the manager gains a value, that degrades with the delay of the packet. The goal is to model the transmission of packets bearing time-sensitive data such as audio or video, where the packet delay is a major parameter in the assessment of the utility it obtains.

Subsequently, we assume that the QoS class of a packet has all the necessary information regarding its value. *I.e.*, it completely specifies the value of sending the packet as a function of its delay.

A queue manager is confronted with two challenges, resulting from informational barriers: (i) Lack of information regarding the future arrival of packets, due to chaotic behavior of incoming flows [99]. (ii) The marking is done voluntarily the users, and selfish users can mark packets improperly to improves their performance.

To handle the first challenge we use *competitive analysis* [18] to evaluate the performance of queue management algorithms, meaning that we derive bounds on the ratio between the total value of the optimal offline scheduling, to the total value of the online policy, over the worst case input. To cope with selfish marking, we associate a cost with the service. Our pricing schemes charge for every packet in a way that encourages users to report their QoS class truthfully.

Competitive analysis has been applied to buffer management of deadline sensitive packets, known as the *bounded delay* model, where the time-loss function is a step function. The initial value from sending a packet drops to 0 after a deadline is reached. Kesselman et al. [69] show that a greedy algorithm, which sends the most valuable packet at every step attains a competitive ratio of exactly 2. In a restricted model, where packets have only two possible values, low value 1, and a high value $\alpha > 1$, they show that the greedy has a competitive ratio of exactly $1 + 1/\alpha$.

Li et al. [74] improve the competitive ratio of the bounded delay problem to 1.854 for the general bounded delay model. And Englert and Westermann [34] showed a 1.893-competitive memoryless algorithm and 1.828 competitive history dependent algorithm. Chin et al. [25] present a randomized algorithm that achieves a competitive ratio of $e/(e-1) \approx 1.582$. Chin and Fung [26] present a lower bound of 5/4 on the competitive ratio of any randomized algorithm. If the deadline of a subsequent packet must be past the deadline of an earlier packet, then Li et al. [73] give an optimal online algorithm with a competitive ratio equal to the golden ratio.

In [69, 7, 74, 34, 25, 26, 73] the algorithm is not informed regarding future arrivals, but it is assumed that the queue manager is fully informed about the type of a packet, for every packet, *i.e.*, a packet's value as a function of time. When every packet represents a selfish agents, wishing to maximize the value from sending the packet, they do not necessarily report the true type of a packet, unless they are incentivized to do so (the algorithms described in those works do not motivate truth telling).

The theory of mechanism design deals with the problem of incentivizing agents to report their type truthfully. When the private type of an agents is a single number, the mechanism design problems is said to be a "single parameter" problem. A lot is known about the design of mechanisms for single parameter agents [9]. In particular, every monotone algorithm, namely, where an increase in the value of a winning agent keeps him winning, admits a truthful mechanism. The payment scheme associated with a monotone algorithm charges every winner the minimum value that would have been sufficient for him in order to win. For example, in the bounded delay problem, when the private type of a packet is its value, and the arrival time and deadline are known to the mechanism, the mechanism design problem is in the single parameter domain. Notice that the greedy algorithm, which is monotone, admits a truthful mechanism.

Cole, Dobzinski, and Fleischer [28] define a special class of "prompt mechanisms", as mechanisms that are able to compute the fee to charge a packet before it is being sent. They observe that the greedy algorithm [69] does not admit a prompt mechanism, and propose a prompt mechanism with a competitive ratio 2. They also show a matching lower bound on every prompt mechanism.

Chapter 6 deals with loss per time which is more moderate than the step function in the bounded delay model. In particular, we focus our attention on the case where the value of a packet degrades linearly in time. Naor [81] studied a linear loss model, in a similar queueing problem. Instead of letting a centralized authority do the admission control, Naor considered a queue formed by selfish agents. In his model, known today as Naor's model, selfish agents arrive at a FIFO queue, inspect the queue size and decide whether to join or balk. Each agent has a benefit of R from a completed service, but there is a fixed cost for every unit of time wasted in waiting for the service. The agents arrival process in Naor's original model is distributed as a Poisson process, and the service time is exponentially distributed[8].

Naor points out that selfish customers will enter the queue even if their benefit from doing so is small, concluding that individuals decisions are not socially optimal. He shows how to compute a socially optimal joining strategy[9] as a function of the arrival rate. To incentivize agents to join in accordance with the optimal strategy, Naor suggests levying tolls on the agents, at the queue entry point, assuming that the utility of an agent to be quasi-linear with money. The appropriate toll, effectively regulates the queue size, and hence achieves (expected) optimal social welfare.

Naor's model had been extensively studied in various setting. Knudsen [71] extends Naor's results to multi-server systems, while simultaneously generalizing the benefit-cost structure. Yechiali [101] extends Naor's result to queues with general (that is, not necessarily poisson distributed) inter-arrival times. Adler and Naor [3] study the case of constant service rate. For a non-FIFO queue management, Hassin [58] shows that optimal social welfare can be attained without tolls in a LIFO queue, when customers can renege at any time. For a detailed discussion see [59].

Competitive Queue Management for Latency Sensitive Packets

The focus of Chapter 6 is online queue management for packets with value that strictly decreases with time. Several versions of this problem are considered.

We first consider a non-preemptive model, where the queue manager cannot drop a packet after it is admitted, and a FIFO queue discipline. The simplest case is that of homogenous packets, where all packets have the same intrinsic value upon arrival, and a constant loss per unit of time. We give a lower bound of $\phi \approx 1.618$ on the competitive

[8]These assumptions correspond to the usual M/M/1 queueing model.

[9]Naor only considers threshold strategies, where the agents join the queue if its size is below a threshold, and otherwise balk.

ratio (even for randomized algorithms), which we match with a simple threshold policy that attains this competitive ratio.

For heterogenous packets and linear value loss,

(i) We give a simple threshold queue policy, “doubling threshold”, with a competitive ratio of 8.

(ii) We observe that this problem has an $O(n \log n)$ time optimal offline algorithm (improving upon the obvious matching approach).

(iii) We show a lower bound of $\phi^3 \approx 4.236$ on the competitive ratio of any deterministic online algorithm.

We also relate the issue of the online competitive ratio to an online mechanism design problem for packets generated by selfish agents. In this case there is little reason to trust the “intrinsic value” claimed by the owner. We reinterpret our online algorithms as yielding an incentive compatible online pricing scheme for heterogenous packets, that guarantees a constant fraction of the optimal social welfare (defined as the sum of agent utilities).

We also consider a model that allows preemption, *i.e.*, dropping packets from the queue, and a queue discipline which is not FIFO. For the homogenous packets case, we show that the LIFO policy attains the optimal solution, and that no pricing mechanism is required to implement this policy. For the case of heterogenous packets we show that the greedy algorithm attains a 2-competitive ratio even if the latency cost function is arbitrary (but non-increasing). In addition, we show a randomized algorithm with a competitive ratio of at most $e/(e-1)$.

Table 1.5 summarizes our results regarding the competitive ratio for different variants of the online queue management problem.

A preliminary version of our results on competitive queue management for latency sensitive agents appears in [40].

Policy	**Value**	**Time cost**	**Lower bound**	**Upper bound**
FIFO	Homogeneous	Homogeneous	$\phi \approx 1.618$; for randomized algorithms	ϕ
FIFO	Heterogeneous	Homogeneous	$\phi^3 \approx 4.236$ for deterministic algorithms	8
LIFO	Heterogeneous	Homogeneous	1	1
Preemptive; deterministic	Heterogeneous	Heterogenous	1	2
Preemptive; randomized	Heterogeneous	Heterogenous	1	$e/(e-1) \approx 1.58$

Table 1.2: Summary of results for latency sensitive queue management, with a linear time cost function.

Chapter 2

Preliminaries

We use $\mathbb{R}$ to denote the real numbers, and $\mathbb{R}_+$ to denote $[0, \infty)$.

2.1 Game Theory

In this section we cover some basic models and equilibrium concepts from game theory which are used in this thesis. We begin with the most basic game model. A game in strategic form is a model for interactive decision making in which every agent makes her decision, and the agents are taking actions simultaneously. An agent's choice comes from a set of strategies; every agent is endowed with a utility function that maps the vector of all agents choices to the reals. Every agent makes her choice so as to maximize her own utility function, and a vector of choices is said to be in a *Nash equilibrium*, if no agent can increase her utility by changing her own action while the other agents stick to their choices.

Definition 2.1. *[Strategic game] A strategic game is a triple $\Gamma = \{N, \{S_i\}_{i\in N}, \{u_i\}_{i\in N}, \}$, that consists of*

(i) A finite set $N = \{1, \dots, n\}$ denotes the set of agents*, where n is a positive integer.*

(ii) For each agent $i \in N$ a nonempty set S_i (the set of actions available to agent i)

(iii) For each agent $i \in N$ a real valued utility function u_i from $S = \prod_{i\in N} S_i$ to the positive reals.

For $s \in S$ let s_{-i} denote the strategy combination of all agents except i, *i.e.*, $s_{-i} = (s_j)_{j\neq i}$. An action profile is said to be in a Nash equilibrium if no agent can increase her payoff by unilaterally deviating.

Definition 2.2. *[Nash equilibrium in pure strategies] A Nash equilibrium in pure strategies of a strategic game is a profile $s^* \in S$ of actions with the property that for every agent i we have*

$$u_i(s_i^*, s_{-i}) \geq u_i(s_i, s_{-i}) \ , \ \textit{for all } s_i \in S_i$$

Thus for s^* to be a Nash equilibrium it must be that no agent i has an action yielding a higher payoff to that generated when she chooses s_i^*, given that every other agent chooses her equilibrium strategy s_j^*.

An action $s_i \in S_i$ is an ϵ-best response to $s_{-i} \in S_{-i}$ if for every action $r_i \in S_i$, $u_i(s_i, s_{-i}) \geq u_i(r_i, s_{-i}) - \epsilon$. We denote by $\mathrm{BR}(s_{-i})$ the set of all 0-best response actions to s_{-i}. A joint strategy $s \in S$ is an *ϵ-Nash equilibrium*, if for each agent i we have that $s_i \in S_i$ is an ϵ-best response to s_{-i}. For $\epsilon = 0$ a 0-Nash equilibrium is a Nash equilibrium.

The notion of a mixed strategy Nash equilibrium models equilibrium of a game in which participants actions are generalized to be probability distributions over their actions set, rather than pure deterministic actions. In this case, a selfish agent acts so as to maximize her expected return. Let $\Delta(S_i)$ denote the set of all probability distributions over S_i, and refer to a member in $\Delta(S_i)$ as a mixed strategy of agent i. A *mixed extension game* of Γ is derived by setting $\Delta(S_i)$ as agent i strategy space, and setting agent i utility function to be her expected utility, over the joint profile of mixed strategies. A mixed strategy Nash equilibrium of a strategic game is a Nash equilibrium of its mixed extension.

A Nash equilibrium in pure strategies need not always exists. However, Nash [82], in his seminal work, shows that a mixed strategy Nash equilibrium, exists for every game.

Concave Games

Rosen [89] extends the definition of a strategic game by considering games where a strategy of an agent is a convex and compact set, and her utility function is concave.

Definition 2.3. *A* concave game *is a game $\Gamma = \{N, \{S_i\}_{i \in N}, \{u_i\}_{i \in N}\}$ such that,*

(i) For every i, the set of actions S_i is a closed, convex and bounded set;

(ii) The utility function u_i of agent i is a strictly concave function in her own arguments. I.e., for every two feasible action profiles (s_i, s_{-i}), and (s'_i, s_{-i}), and every $\lambda \in [0, 1]$,

$$u_i(\lambda \cdot s_i + (1 - \lambda)s'_i, s_{-i}) > \lambda u_i(s_i, s_{-i}) + (1 - \lambda)u_i(s'_i, s_{-i})$$

Rosen shows that a Nash equilibrium always exists in concave games. Furthermore, in case that the utility functions satisfy an additional concavity requirement, he calls *diagonal strict concavity*, the equilibrium of a concave game is unique. To explain the diagonal strict concavity requirement, we define a function $\sigma : \prod_{i\in N} S_i \times \mathbb{R}^n \to \mathbb{R}$ to be a non-negative sum of the utility functions, $\sigma(s,\lambda) = \sum_{i\in N} \lambda_i u_i(s)$. For each fixed $\lambda \in \mathbb{R}^n$ a related mapping $g(s,\lambda)$ in terms of the gradients $\nabla_i u_i(s)$[1] is defined by

$$g(s,\lambda) = \begin{bmatrix} \lambda_1 \nabla_1 u_1(s) \\ \lambda_2 \nabla_2 u_2(s) \\ \vdots \\ \lambda_n \nabla_n u_n(s) \end{bmatrix}. \tag{2.1}$$

Definition 2.4 (Diagonal strict concavity from [89])**.** *The function σ will be called diagonally strictly concave for a joint strategy space S, and a fixed $r \geq 0$, if for every two distinct strategy profiles, $s^0, s^1 \in S$, we have*

$$(s^1 - s^0)^T g(s^0,\lambda) + (s^0 - s^1)^T g(s^1,\lambda) > 0$$

Theorem 2.5 (from [89])**.** *Let Γ be a concave game for which the diagonal strict concavity property is satisfied. Then Γ admits a unique Nash equilibrium point.*

2.2 Multi-Stage Games

In many situations agents engage in the same interaction over again and again. For example, consider a set of agents, each of which is required to route the same amount of traffic every second over a common network, or buy a a share of network bandwidth at the beginning of every day. In such an interaction agents receive a payoff after every stage of the game, and can change their actions from stage to stage based on feedback they received in previous stages.

A *multi-stage game* is a basic model for this type of interaction, that consists of repeatedly, and simultaneously playing a one shot game $\Gamma = \{N, \{S_i\}_{i\in N}, \{u_i\}_{i\in N}\}$. Namely, on every stage $t = 1, 2, \ldots$, every agent $i \in N$ simultaneously chooses an action, $s_i^t \in S_i$, and

[1]The i'th element of the gradient $\nabla_i u_i(s)$ is defined as the gradient of u_i in the i'th strategy, *i.e.*, when $s_i \in \mathbb{R}^{m_i}$, $m_i > 0$ is a positive integer, then $\nabla_i(u_i(s)) = (\frac{\partial}{\partial s_i^1} u_i(s), \ldots, \frac{\partial}{\partial s_i^{m_i}} u_i(s))$.

subsequently receives a utility $u_i(s^t)$. A strategy of an agent in a multi-stage game maps every finite history of the game to her actions set in the one shot game.

Multi-stage games are important in both game theory, and learning theory. From a game theoretic view point they allow modeling the fact that when an agent takes some action, there may be consequences for this beyond the immediate payoff. From a learning perspective, repetition in games gives rise to a simple form of learning from past experience.

Regret Minimization

Regret minimization arises in the context of repeatedly playing a game against nature or other agents. The *external regret* of an individual agent compares in retrospect, the agent's average utility to that of the best static procedure in hindsight.

To formally define external regret, consider Alice who interact repeatedly in a game against an adversary. At every stage $t = 1, 2, \ldots$, Alice chooses an action a^t from a set S of K alternatives. The adversary chooses a cost $\ell^t(a)$ for every action $a \in S$. Alice's external regret at time T is defined as

$$\left(\sum_{t=1}^{T} \ell^t(a^t)\right) - \min_{a \in S} \left(\sum_{t=1}^{T} \ell^t(a)\right).$$

In a full information setting, Alice receives as feedback the complete loss vector $(\ell^t(a))_{a \in S}$, after choosing her action a^t. In a *multi-armed bandit* setting, Alice is revealed with her own cost at time t, $\ell^t(a^t)$, but not with the entire cost vector.

In the full-information setting, the optimal regret bound is $O(T \log(K))$, due to Freund and Schapire [48].

In the context of multi-agent games, where utilities rather than costs are considered, the *external regret* of an agent $i \in N$ at a time T, is

$$\max_{r_i \in S_i} \left(\sum_{t=1}^{T} u_i(r_i, s^t_{-i})\right) - \left(\sum_{t=1}^{T} u_i(s^t)\right),$$

where s^t is the vector of actions played at time t.

Online Concave Optimization

An online concave[2] optimization problem, models an online decision making process with the property that the structure of the reward, as a function of the decision variables is concave. Let us concentrate on the case of a single agent. Formally, an *online concave programming* problem, C, consists of a convex feasible set $F \subset \mathbb{R}^d$ and an infinite sequence $\{f^1, f^2, \ldots\}$, where each $f^\tau : F \to \mathbb{R}$ is a concave function. A problem is said to be an online linear optimization problem, if the cost functions are linear.

An *online concave programming algorithm* selects, at each time step t, a vector $x^t \in F$, given a history . After x^t is selected, the algorithm observes the payoff function $f^t(\cdot)$, and receives a payoff of $f^t(x^t)$.

An algorithm for choosing a point in F at every time step is said to have no-external-regret, if the difference between its average performance and that of the best single action in hindsight, diminishes over time. Formally,

Definition 2.6. *Let A be an online concave programming algorithm, then the regret of algorithm A is defined by*

$$\mathcal{R}_A(T) = \max_x \left(\sum_{t=1}^{T} f^t(x) \right) - \left(\sum_{t=1}^{T} f^t(x^t) \right).$$

An algorithm A has no external regret, if $\mathcal{R}_A(T) = o(T)$.

The first efficient algorithm for a linear optimization problem is due to Hannan [52], and later Kalai and Vempala [64]. It achieves a regret bound of $O(\sqrt{T}\log(d))$, which essentially the best possible.

For the general concave case, Zinkevich [103] provides a simple procedure that guarantees a regret bound of $O(\sqrt{T})$, when F is a closed, convex, bounded and non-empty set, and that every f^t is differentiable with a bounded first derivative. Hazan et al. [60], give algorithms that guarantee an upper bound of $O(\log(T))$ on the external regret, when f^t are strictly concave functions and twice differentiable.

[2]Such problems are better known as online convex optimization, with loss functions, rather than reward. As we are more interested in modeling online decisions that results in rewards, we changed convex to concave.

Multi-Armed Bandits

Regret minimization has also been studied in the multi-armed bandit setting, where at each stage t, only the payoff $f^t(x^t)$ is revealed instead of f^t. Auer et al. [11] study a multi-armed bandit problem with a finite number of actions. They present an algorithm that guarantees that the external regret at time T, is at most $O(\sqrt{KT \ln K})$, when the decision maker has K actions to choose from, and supplement it with an almost matching lower bound of $\Omega(\sqrt{KT})$ on the external regret of every algorithm.

Awerbuch and Kleinberg [13] have initiated the study of non-stochastic multi-armed bandit problems in the context of online linear optimization problems. They provide an algorithm with a regret bound of $O(\text{poly}(d)T^{2/3})$. Flaxman et al. [41] show how to obtain a regret bound of $O(T^{3/4})$ in an online concave problem. Abernethy et al. [1] show an efficient algorithm that archives a regret bound of $O(T^{1/2})$ for the online linear optimization problem.

2.3 Competitive Analysis in Online Decision Problems

In an online decision problem, the items that compose of a sequential decision problem are revealed sequentially, and a decision maker is required to make a decision after every piece of information is revealed. These decisions will have an impact on the quality of her overall performance.

In many situations a decision must be made in the absence of some important information. For example, imagine a factory getting manufacturing orders along with a price quote and must accept or reject every order on spot. If the factory is already committed for doing a certain set of orders, he might not be able to accept an attractive offer with a much higher price quote. But on the other hand, if the factory decides to reject an order, for wanting to keep availability for better offers to come, it may be left with nothing, if new orders fail to arrive.

One way to model uncertainty is the Bayesian approach, where uncertainty is modeled as a prior probability distribution over the possible outcomes. A decision maker then optimizes her expected utility. In a network setting, priors are often hard obtain, due to the sometime chaotic nature of data transportation. Thus, in contrast with the Bayesian approach, we take the *competitive analysis* approach (cf. [18]), where no prior is available for the decision maker; actions must be made solely based on observations from past events. The performance of an online algorithm is measured using its competitive ratio defined as the worst case ratio between the performance of the online algorithm, and that of an off-line algorithm, which is given all the information in advance.

To formally define the notion of competitive ratio, we first define an optimization problem P to consists of an Input I, a set of feasible output solutions $F(I)$, and for every feasible output $f \in F(I)$, a reward $U(I, f)$. Given any legal input I, an algorithm ALG for P computes a feasible output ALG$[I] \in F(I)$. The reward associated with this feasible output is denoted by $ALG(I)$. An optimal algorithm OPT is such that for all legal inputs,

$$\mathrm{OPT}(I) = \max_{o \in F(I)} U(I, o).$$

In an online problem the input is revealed in an online manner, and the output must be produced online. An algorithm ALG is required to decide on the output immediately, based only on the information revealed thus far, and on previous decisions.

Algorithm ALG is *c-competitive*, if there is a constant α such that for all finite input sequences I,

$$\mathrm{ALG}(I) \geq c \cdot \mathrm{OPT}(I) + \alpha.$$

Chapter 3

Strategic Protocols for Collision Resolution

Ethernet buses and wireless communications are both examples of shared communication media. Transmission is successful on such channels only if exactly one user accesses the media. Should multiple users access the channel simultaneously, a collision is said to occur, and all attempted transmissions fail. Contention resolution protocols are designed to address the problem of collisions, and to ensure fair and efficient use of such channels.

One would like to have a distributed contention resolution protocol, where anonymous users know little, if anything, about others. The celebrated Aloha protocol is an excellent example of such a distributed contention resolution protocol. Since the introduction of the Aloha protocol, much research has been devoted in deriving improved contention resolution protocols, where the main emphasis has been the stability of the protocol at high loads. (See [88] for an excellent treatment of the topic.)

Assume n agents at time zero, each with one packet to transmit. Agents that transmit without collision on the channel are said to be *successful*. A successful agent departs and is no longer in contention for the channel. If at every time slot, each of the k yet unsuccessful agents transmits with probability $1/k$, then the expected latency per agent is overall $\Theta(n)$, and with high probability no agent will be unsuccessful after $10n$ time slots. For symmetric protocols this is the *socially optimal protocol* in terms of minimizing the expected sum of latencies, expected maximum, etc [1].

[1] More precisely, this protocol is the optimal history independent protocol, however, there exist protocols that achieve better throughput based on the channel's history (see [88]). As the $1/k$ transmission probability

In this chapter we study contention resolution in the context of selfish user behavior. In the problems we study, rational and selfish agents seek to minimize their own costs, and have no compulsion to avoid harming others. One simple example of an agent's cost function is the *latency cost*, where the cost of a packet is the time delay between packet creation and successful packet transmission. Unless stated explicitly otherwise, we consider latency costs hereinafter.

Rational selfish agents with latency costs will subvert the socially optimal protocol given above. Consider Alice who, starting at time zero, continuously transmits until successful. If the other agents follow protocol and transmit with probability $1/n$ while Alice is still unsuccessful, then the expected latency for Alice drops from $\Theta(n)$ to $O(1)$.

One can view the problem of devising protocols for selfish agents as a problem in mechanism design. However, we stress that we only allow protocols that are self-enforceable and do not involve external payments or incentives. In fact, we view protocol design here as searching for "good" equilibria, with seemingly surprising results. Using the terminology of [72], what we prove here is that the price of anarchy for contention resolution games is infinite, whereas the price of stability[2] for contention resolution games is $O(1)$.

A priori, one might naturally suspect an impossibility result, that all agents will continuously transmit, and therefore that no success will ever occur. Many examples of such selfish behavior have been shown in the game theory literature, this includes the prisoners dilemma and the "tragedy of the commons" (see [53]), where agents need cooperation to profit from a common resource.

In our setting, if the number of agents is at least three, then continuous transmission by all agents is indeed in equilibrium. In this natural equilibrium no transmission is ever successful. We call such an equilibrium a blocking equilibrium. However, there exist other *non-blocking* equilibria, where — eventually — all agents succeed. To see why this is so, consider two agents, Alice and Bob, each with one packet to send, and both seeking to minimize latency cost (delay until packet transmission). If Alice chooses to be aggressive and broadcast endlessly until successful transmission, the best response for Bob is to allow Alice to transmit in the first time slot, following which Alice loses interest in competing with Bob, and Bob now has full access to the channel without further interference.

We generalize the Alice and Bob example above to multiple agents, Alice, Bob, Carol,

attains expected linear delay, we use it as a benchmark and regard it as the socially optimal protocol.

[2]With one major caveat, that this only holds with high probability.

. . ., and again — assume one packet per agent. A strategic agent, Bob, will somehow have to balance the following profit/loss outcomes as influenced by his actions:

1. Immediate success, if Bob choose to transmit and none other did so, or

2. Delayed gratification, if Bob refrained from transmission and some other agent was successful. Bob has gained because there are now less agents in contention for the shared media, or

3. "Wasted" time slots: either collisions or no transmissions on the channel, neither Bob nor any other agent is successful.

It may be illuminating to contrast our game with the repeated Prisoners Dilemma. In the finite horizon repeated Prisoners Dilemma, defecting is always the right choice. In our contention resolution setting this is not true. In a repeated game, the next game is exactly the same as the current game irrespective of the outcome of the current game. In our setting of a simultaneous play exhaustive form game, the game to be played next depends on the outcome of the current game. Fortunately, this gives non-trivial and socially desirable equilibria for various utility functions, even given fixed predetermined horizons.

Furthermore, it seems that the "folk theorems" [85] about cooperation and punishment in repeated games are not directly applicable to our problem. In the folk theorems, a misbehaving agent may gain momentarily, but will receive punishment soon thereafter. In out setting, a defecting agent may succeed in attaining his ultimate goal via defection (for example, hogging the transmission channel until successful transmission). Unlike a repeated game, defectors who have concluded their affairs will not hang about to receive punishment. This said, our efficient protocols in equilibrium do include the introduction of fear from overhanging "global" disaster that induces what seems to be cooperation.

We deal with synchronous communications where transmissions are only possible in discrete time slots. We assume that every agent has one packet to transmit. Additionally, we deal with symmetric protocols, where all agents follow the same set of rules[3]. Our

[3] In applications such as mobile applications over broadcast channels, anonymity occurs naturally and may even be a requirement. In such settings, it makes little sense to consider non-symmetric protocols where the three agents Alice, Bob, and Carol, each play a different strategy depending on their own identity and the identities of others with which they are playing. If we were to allow known (and unique) identities then the contention resolution problem becomes somewhat uninteresting. One could use social rank to determine priorities based upon identities, and this result is in equilibrium (possibly considered unfair by those of lower social standing).

primary interest is in latency costs as agent utilities, but we also study the effect of a deadline.

We study the effect of time dependency on protocols. Time-independent protocols may determine transmission probability using the current number of agents in contention (called pending agents), but may not use the current time slot index. A time-dependent protocol is not so restricted. For example, the socially optimal protocol (transmission probability $1/k$ for k pending agents) is a time independent protocol but is not in equilibrium.

We show that there is a unique time-independent non-blocking symmetric protocol in equilibrium, in which all agents broadcast with probability $p_k \in \Theta(1/\sqrt{k})$ (again — k is the number of pending agents). With such transmission probabilities, the expected duration until all n agents succeed is approximately $e^{\Theta(\sqrt{n})}$ (which is dominated by the expected latency of the first successful transmission). We deal with time independent protocols for latency in Section 3.3.

We define a protocol to be *efficient* if the maximal packet latency is linear with high probability. The socially optimal protocol (which sends with probability $1/k$) is efficient but is not in equilibria. Broadcasting with probability $p_k \in \Theta(1/\sqrt{k})$ is in equilibria but is not efficient. Furthermore, our claims above imply that any symmetric protocol that is simultaneously efficient and in equilibria must be time-dependent. Thus, we seek an efficient time-dependent protocol in equilibrium for latency costs. This motivates our study of deadline cost functions and suggests the notion of virtual deadlines, which we can use to derive efficient protocols.

A deadline cost function would typically charge only those agents that have not been successful prior to the deadline. *E.g.*, a tax audit for those not filing by midnight. Perhaps surprisingly, one sees dramatic behavioral changes in equilibria as a function of the time left until the deadline. If the deadline is close by (say $2n$ time slots away), then the only equilibria for selfish agents is to transmit with high probability (and thus the probability that *any* agent will be successful is negligible). Given a deadline $15n$ time units away, then — with very high probability — all n agents will succeed prior to the deadline. Deadline cost functions can be used to model Quality of Service issues, *e.g.*, MPEG packet delivery past a deadline causes video breakup. Section 3.4 studies protocols for agents with deadline cost functions (as well as agents with latency costs).

We seek equilibria[4] where "ill behaved" latency cost agents behave more like "polite"

[4] We remark that our protocol is not only in equilibria but also subgame perfect.

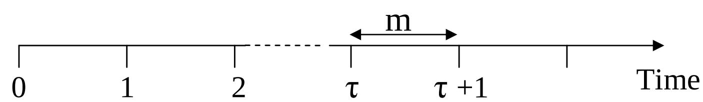


Figure 3.1: At the beginning of every time slot $t \geq 0$ the agents choose whether to transmits or be quiescent. Even if a transmission is successful, it still takes the message m one time slot to be transmitted; if Alice successfully transmits at time slot τ, then her latency is $\tau + 1$, and her cost is $\Psi(\tau + 1)$.

deadline cost agents, for an appropriately chosen deadline. We stress again that we are not introducing external payments or charges to introduce the deadline, and we are not changing the latency cost assumption about the agents. Our protocol is "self policing" and enforces a "virtual" deadline on the agents, of sufficiently great cost so that they transmit with probability $O(1/k)$.

3.1 Preliminaries

We consider the following contention resolution problem. Consider a set of n agents, each of which has a *single* packet to transmit. Agents that have not yet successfully transmitted their packet are called *pending*, initially all n agents are pending.

Time consists of discrete time slots. Agents that are pending at time slot t can either *'Transmit'* or be *'Quiescent'*. If exactly one agent chooses to transmit at time slot t then this agent is *successful* and ceases to be pending. If multiple agents choose to transmit at time t then a *collision* occurs. In case of collision or if the channel is idle then the set of pending agents remains unchanged. The number of agents at time zero, n, is known to all agents, and the agents keep track of K_t — the number of pending agents at time t.

We study multiple agent access to a channel as a non-cooperative game in extensive form and simultaneous play. The latency T_i for agent $i \in \{1, \ldots, n\}$ is a random variable whose value is the time at which agent i is successful (or ∞), and whose distribution is determined by the (possibly) mixed strategies of the agents. The transmission of a message takes the duration of a time slot, thus an agent i that transmits successfully at time slot t, has a latency $T_i = t + 1$ (*E.g.*, if agent Alice transmits at time $t = 0$, and she is successful, then her latency $T_{\text{Alice}} = 1$, due to the time it takes the message to be transmitted). The cost to agent i is a function of the latency, $\Psi(T_i)$, and is thus also a random variable.

Our primary interest is in the latency cost function for all agents, *i.e.*, $\Psi(t) = t$. We also present results for deadline cost functions (*e.g.*, $\Psi(t) = 0$ for $t < D$ and $\Psi(t) = M$ for $t \geq D$). Section 3.5 deals with exponentially increasing/ decreasing marginal cost functions *i.e.*, $\Psi(t) = \sum_{j=0}^{t-1} \delta^j$, for some constant δ.

Definition 3.1. *A strategy for agent Alice, $q = \langle q_{k,t} : 1 \leq k \leq n, 0 \leq t \rangle$, is interpreted as follows: if Alice is one of k pending agents at time t (i.e., $K_t = k$), then Alice transmits with probability $q_{k,t}$.*

A strategy for agent Alice is said to be time-independent *if the transmission probabilities, $q_{k,t}$, are independent of the time,* i.e., *$q_{k,t} = q_{k,t'}$, for all $0 \leq t, t'$. A time-independent strategy can thus be represented as a vector $q = \langle q_1, q_2, \ldots, q_n \rangle$, where q_k is the transmission probability given k pending agents, irrespective of the time*[5].

Definition 3.2. *A* protocol *$Q = \langle q^{(1)}, q^{(2)}, \ldots, q^{(n)} \rangle$ is a list of strategies, one per agent, where agent i, $1 \leq i \leq n$ has strategy $q^{(i)}$.*

Fix a protocol Q. We define the expected cost of the protocol for agent i, as $C_i^Q = \mathrm{E}[\Psi(T_i)]$, where the expectation is taken over the probability distribution defined by Q. Let $T_{i|k,t}$ denote the random variable of the latency for agent i, conditioned on $K_t = k$, and on agent i being one of the k pending agents. Let $C_{i|k,t}^Q = \mathrm{E}[\Psi(T_{i|k,t})]$, and define the expected future cost $F_{i|k,t}^Q = C_{i|k,t}^Q - \Psi(t) = \mathrm{E}[\Psi(T_{i|k,t})] - \Psi(t)$. (When clear from the context we drop the superscript Q.)

Definition 3.3. *Let $Q = \langle q^{(1)}, \ldots, q^{(n)} \rangle$ be a protocol. Let (s, Q^{-i}) denote the protocol where agents $j \neq i$ use strategies $q^{(j)}$ and agent i uses strategy s. We say that strategy s is a* best response *of agent i to Q^{-i}, if the expected cost to i with s, given that other agents $j \neq i$ use $q^{(j)}$, is minimal. I.e., s is a best response to Q^{-i} if for all strategies r,*

$$C_i^{(s,Q^{-i})} \leq C_i^{(r,Q^{-i})}.$$

We say that protocol Q is in equilibria if $q^{(i)}$ is a best response to Q^{-i} for all agents i.

For one pending agent, the best response for the agent is to transmit deterministically. *I.e.*, for protocols Q in equilibrium, $q_{1,t} = 1$, for all $t \geq 0$. Consequently, $T_{i|1,t} = t + 1$ and $C_{i|1,t} = \Psi(t + 1)$.

[5] *For latency costs, classical Markov Decision Theory results [87] show that the best response to a set of time-independent strategies will include some time-independent strategy.*

Definition 3.4. *A protocol Q is said to be symmetric if $q^{(i)} = q^{(j)}$, for all $i, j \in N$. For symmetric protocols one can use the notation $Q = \langle q \rangle^n$ rather than $Q = \langle q^{(1)}, \ldots, q^{(n)} \rangle$. For the expected cost to an agent we use the notation $C^Q_{k,t}$ instead of $C^Q_{i|k,t}$, as the index i is irrelevant. Likewise, the cost of the protocol can be denoted by C^Q in place of $C^Q_{i|n,0}$.*

For $k \geq 3$, having all the agents continuously transmit (*i.e.*, $q_{k,t} = 1$) is a symmetric, time-independent, protocol in equilibria. Such a protocol is also rather useless as no successful transmissions ever occur[6].

Definition 3.5. *A protocol is called* non-blocking *if for all $k \geq 2$, $t \geq 0$, the transmission probability $q_{k,t} < 1$.*

Note that the expected cost of the game for a time-independent, non-blocking protocol in equilibria is always finite (for the latency cost).

Definition 3.6. *Let $Q = \langle q^{(1)}, \ldots, q^{(n)} \rangle$ be a protocol. Q is said to be efficient if all agents are successful within $D = O(n)$ time slots, except possibly with exponentially negligible probability ($1/\exp(n)$).*

It does not follow from the definition of efficient protocols that the *expected cost of the game* need be low. Of course, this depends on Ψ, but even for latency costs, efficient protocols could have very high latency with some (exponentially small) probability and the expected latency could also be high.

3.2 Characterization of Symmetric Protocols in Equilibria

In this section we analyze properties of symmetric protocols in equilibria, for general non-negative cost functions. For any symmetric protocol, $Q = \langle q \rangle^n$, where $q = \langle q_{k,t} : 1 \leq k \leq n, 0 \leq t \rangle$, the expected cost for any agent (*e.g.*, Alice), conditioned on the event that k agents, Alice amongst them, are pending at time t, is

$$\begin{aligned} C_{k,t} &= q_{k,t}(1 - q_{k,t})^{k-1}\Psi(t+1) \\ &\quad + (k-1)q_{k,t}(1 - q_{k,t})^{k-1}C_{k-1,t+1} \\ &\quad + (1 - kq_{k,t}(1 - q_{k,t})^{k-1})C_{k,t+1}. \end{aligned}$$

[6] As mentioned in the introduction, for two agents this is not an equilibrium, and $q_{2,t} < 1$.

Expression	**Definition**
$\Psi(t)$	The cost for an agent who leaves at time t. I.e., successful transmits at time slot $t-1$.
$\psi(t)$	The marginal cost at time t, *i.e.*, $\psi(t) = \Psi(t) - \Psi(t-1)$.
$q_{k,t}$	Transmission probability when k agents are pending at time t.
q_k	Transmission probability of a time-independent protocol when k agents are pending.
q	A strategy vector; a time independent strategy is a vector $q = \langle q_1, \ldots, q_k \rangle$; a time dependent strategy is a matrix $q = \{q_{k,t}\}_{1 \leq k \leq n, 0 \leq t}$.
$\overline{q}^{k,t}$	A strategy vector equal to q except that $\overline{q}_{k,t} = 1$.
$\underline{q}^{k,t}$	A strategy vector equal to q except that $\underline{q}_{k,t} = 0$.
Q	A protocol, consists of n strategies. $Q = \langle q^{(1)}, q^{(2)}, \ldots, q^{(n)} \rangle$.
Q^{-i}	A vector of $n-1$ strategies, omitting the i'th strategy $Q^{-i} = \langle q^{(1)}, \ldots, q^{(i-1)}, q^{(i+1)}, \ldots, q^{(n)} \rangle$
$\alpha_{k,t}$	The probability that none of $k-1$ pending agents transmit at time t.
$\beta_{k,t}$	The probability that exactly one of $k-1$ pending agents transmits at time t.
$C_{k,t}$	The expected cost to one of k pending agents at time t.
$F_{k,t}$	The expected future cost to one of k pending agents at time t, *i.e.*, $F_{k,t} = C_{k,t} - \Psi(t)$.
K_t	A random variable indicating the number of pending agents at time t. K_t is distributed in accordance with the protocol Q in use.
T_i	The latency of agent i, i.e., the time slot number at which i successfully transmits + 1.

Table 3.1: Notation in use.

The first term above is the contribution to the expected cost conditioned on Alice successfully transmitting at time slot t. The second term is the contribution conditioned on some other agent (not Alice) transmitting successfully. The last term is the contribution to the expected cost when there is no successful transmission (either no agent attempts transmission or multiple agents attempt transmission).

For an agent strategy q, the strategy $\overline{q}^{(k,t)}$ (respectively, $\underline{q}^{(k,t)}$) is the same as q except that it deterministically transmits (respectively, is quiescent) at time t if $K_t = k$, *i.e.*, $\overline{q}^{(k,t)} = q$ (respectively, $\underline{q}^{(k,t)} = q$) except that $\overline{q}^{(k,t)}_{k,t} = 1$ (respectively, $\underline{q}^{(k,t)}_{k,t} = 0$).

Given that $K_t = k$, the expected cost to Alice, playing strategy $\overline{q}^{(k,t)}$, is

$$C_i^{(\overline{q}^{(k,t)},Q^{-i})} = \alpha_{k,t}\Psi(t+1) + (1-\alpha_{k,t})C_{k,t+1},$$

where

$$\alpha_{k,t} = (1-q_{k,t})^{k-1}\ , \tag{3.1}$$

is the probability that none of the other $k-1$ pending agents transmit at time t. Similarly, the expected cost to Alice when playing $\underline{q}^{(k,t)}$ is

$$C_i^{(\underline{q}^{(k,t)},Q^{-i})} = \beta_{k,t}C_{k-1,t+1} + (1-\beta_{k,t})C_{k,t+1},$$

where

$$\beta_{k,t} = (k-1)q_{k,t}(1-q_{k,t})^{k-2} \tag{3.2}$$

is the probability that exactly one (other) pending agent transmits.

For protocols Q in equilibria, for any k, t such that $0 < q_{k,t} < 1$, it must be that all three strategies, q, $\overline{q}^{k,t}$, and $\underline{q}^{k,t}$, are best responses to Q^{-i}.[7]

We now argue that for symmetric protocols in equilibria, expected cost is monotonically increasing in the number of pending agents and in time.

Lemma 3.7. *Let $Q = \langle q\rangle^n$ be a symmetric protocol in equilibria. For all $k \le n$ and all $t \ge 0$,*

$$C_{k,t} \le C_{k,t+1}, \quad C_{k-1,t} \le C_{k,t}, \quad \text{and } F_{k-1,t} \le F_{k,t}.$$

Proof. Consider the case of k pending agents, one of which is agent i. Recall that in equilibrium, each agent has the same expected cost for playing either q or $\overline{q}^{(k,t)}$ or $\underline{q}^{(k,t)}$.

[7] This follows since both Transmit and being Quiescent are in the support of q at time t with k pending agents.

Namely,

$$\begin{aligned} C_{k,t} &= \alpha_{k,t}\Psi(t+1) + (1-\alpha_{k,t})C_{k,t+1} \\ &= \beta_{k,t}C_{k-1,t+1} + (1-\beta_{k,t})C_{k,t+1}, \end{aligned}$$

where $\alpha_{k,t}, \beta_{k,t}$ are as defined in equations (3.1), and (3.2) respectively.

All agents pending at time $t+1$ will not have success prior to time $t+2$ and thus have cost $\geq \Psi(t+2)$, *i.e.*,

$$C_{k,t+1} \geq \Psi(t+2) \geq \Psi(t+1) \ .$$

This implies that,

$$C_{k,t} = \alpha_{k,t}\Psi(t+1) + (1-\alpha_{k,t})C_{k,t+1} \leq C_{k,t+1} \ ,$$

establishing that $C_{k,t} \leq C_{k,t+1}$.

By the fact that $C_{k,t} \leq C_{k,t+1}$, we get that

$$C_{k,t} \geq \beta_{k,t}C_{k-1,t+1} + (1-\beta_{k,t})C_{k,t}$$

Hence,

$$\beta_{k,t}C_{k,t} \geq \beta_{k,t}C_{k-1,t+1} \geq \beta_{k,t}C_{k-1,t} \ , \tag{3.3}$$

where the second inequality in (3.3) follows since $C_{k-1,t+1} \geq C_{k-1,t}$. This establishes that $C_{k-1,t} \leq C_{k,t}$.

By definition of $F_{k,t}$:

$$F_{k,t} = C_{k,t} - \Psi(t) \geq C_{k-1,t} - \Psi(t) = F_{k-1,t} \ ,$$

showing that $F_{k-1,t} \leq F_{k,t}$. ∎

The following lemma establishes a connection between the transmission probability $q_{k,t}$, and the ratio of future costs $F_{k-1,t+1}/F_{k,t+1}$.

Lemma 3.8. *Let $Q = \langle q \rangle^n$, be a symmetric protocol in equilibrium. For every number of* *pending agents* $1 \leq k \leq n$ *and for every time slot* $t \geq 0$*, we have*

$$\textit{either} \;\; q_{k,t} = \frac{1}{k-(k-1)\frac{F_{k-1,t+1}}{F_{k,t+1}}} \;, \textit{or} \;\; q_{k,t} = 1 \;.$$

Proof. Consider Alice, one of k pending agents at time t. All agents but Alice follow strategy q. For symmetric protocols, transmission probabilities $q_{k,t}$ are always strictly positive. If $q_{k,t} = 1$ then we are done. Otherwise, in equilibrium, all agents, including Alice, have the same expected cost whether playing q, $\overline{q}^{(k,t)}$ or $\underline{q}^{(k,t)}$. *I.e.*,

$$C_{k,t} \;\; = \;\; \alpha_{k,t}\Psi(t+1) + (1-\alpha_{k,t})C_{k,t+1} = \beta_{k,t}C_{k-1,t+1} + (1-\beta_{k,t})C_{k,t+1}$$

Therefore,

$$\alpha_{k,t}(\Psi(t+1) - C_{k,t+1}) = \beta_{k,t}(C_{k-1,t+1} - C_{k,t+1}).$$

Substituting according to equations (3.1),and (3.2), we get that

$$(1-q_{k,t})^{k-1}(\Psi(t+1) - C_{k,t+1}) = (k-1)q_{k,t}(1-q_{k,t})^{k-2}(C_{k-1,t+1} - C_{k,t+1}).$$

Since by assumption $q_{k,t} \neq 1$, dividing by $(1-q_{k,t})^{k-2}$ results in

$$(1-q_{k,t})(\Psi(t+1) - C_{k,t+1}) = (k-1)q_{k,t}(C_{k-1,t+1} - \Psi(t+1) + \Psi(t+1) - C_{k,t+1}).$$

As $F_{k,t+1} = C_{k,t+1} - \Psi(t+1)$ it follows that

$$(1-q_{k,t})F_{k,t+1} = (k-1)q_{k,t}(F_{k,t+1} - F_{k-1,t+1}),$$

and the claim follows. ∎

Remark 3.9. *In the proof of Lemma 3.8, we implicitly assume that* $C_{k,t}$ *is finite. This holds for any non-blocking protocol with respect to the latency cost function.*

The following are immediate consequences of Lemma 3.8.

Corollary 3.10. *If* Q *is a symmetric protocol in equilibrium then* $q_{k,t} \geq \frac{1}{k}$ *for every integer* $1 \leq k$*,* $0 \leq t$*.*

Corollary 3.11. *For any constant* $0 \leq c < 1$*, if* $F_{k-1,t+1}/F_{k,t+1} < c$*, for all* $k \leq n$*,* $0 \leq t$*, then* $q_{k,t} = \Theta(\frac{1}{k})$*.*

This then implies that the expected latency and the expected maximal latency are both $\Theta(n)$, given that n agents start at time zero.

3.3 Non-blocking protocols in equilibrium for latency cost

Recall that time-independent protocols are protocols in which $q_{k,t} = q_{k,t'}$ for $t \neq t'$. Thus, for such protocols we can use the notation q_k for transmission probability rather than $q_{k,t}$. When time-independent strategies are used for the latency cost function, the future cost depends only on the number of pending agents, *i.e.*, $F_{k,t} = F_{k,t'}$ for $t \neq t'$. We can therefore use the notation F_k for future cost rather than $F_{k,t}$. We give the following characterization of time-independent, non-blocking protocols, for agents with latency costs.

Theorem 3.12. *There is a unique time-independent, symmetric, non-blocking protocol $\langle q \rangle^n$ in equilibrium for latency cost, $q = \langle q_1, \ldots, q_n \rangle$. Furthermore, $q_k \in \Theta(\frac{1}{\sqrt{k}})$, for $1 \leq k \leq n$.*

For the proof of Theorem 3.12 consider agent Alice, one of k pending agents at time t. Assume Alice deviates from q and continuously transmits until successful. This pure strategy is in the support of q, therefore it has an expected cost equal to that of q. As Alice continuously transmits, no agent other than Alice can succeed while Alice is pending. The probability that all agents but Alice are quiescent is denoted $\alpha_k = (1 - q_k)^{k-1}$. Alice's latency is $t+X$, where X is geometrically distributed with parameter α_k, from which follows that the expected number of additional time slots until Alice succeeds $E[X] = 1/\alpha_k$, and

$$F_k = \frac{1}{(1 - q_k)^{k-1}}. \tag{3.4}$$

From Lemma 3.8 and equation (3.4) we have,

$$F_{k-1} = \frac{1}{(1 - q_k)^{k-1}} \left(1 - \frac{1 - q_k}{(k-1)q_k}\right). \tag{3.5}$$

Now, substituting $k - 1$ for k in equation (3.4), we have that $F_{k-1} = \frac{1}{(1-q_{k-1})^{k-2}}$. Substituting this for the left hand side of equation (3.5) we get that

$$\frac{1}{(1 - q_{k-1})^{k-2}} = \frac{1}{(1 - q_k)^{k-1}} \left(1 - \frac{1 - q_k}{(k-1)q_k}\right). \tag{3.6}$$

We first seek to prove inductively that q_k is uniquely determined. For the base case,

$k = 1$, we have that $q_1 = 1$ and $F_{1,t} = 1$ for all $t \geq 0$. Assume the inductive hypothesis for $k-1$, $k > 1$, then the left hand side of equation (3.6) is some constant. The right hand side of equation (3.6) is a continuous and monotonically increasing function of q_k in the range (0,1) taking values from $-\infty$ to ∞. It follows that q_k is uniquely determined.

The proof of the second part of the theorem goes by induction on k. We show how to prove $\ell_k \leq q_k \leq u_k$ for general lower and upper bounds ℓ_k, u_k.

To show that $\ell_k \leq q_k \leq u_k$, for all k, it suffices to show that

1. If $\ell_k \leq q_k \leq u_k$ then $\frac{1}{(1-\ell_k)^{k-1}} \leq F_k \leq \frac{1}{(1-u_k)^{k-1}}$.
2. If $\frac{1}{(1-\ell_{k-1})^{k-2}} \leq F_{k-1} \leq \frac{1}{(1-u_{k-1})^{k-2}}$ then $\ell_k \leq q_k \leq u_k$.

Proposition 3.13. *If $q_k \geq \ell_k$ then $F_k \geq \frac{1}{(1-\ell_k)^{k-1}}$. If $q_k \leq u_k$ then $F_k \leq \frac{1}{(1-u_k)^{k-1}}$.*

Proof. The claim follows directly from equation (3.4), since F_k is monotonically increasing as a function of the transmission probability q_k. ∎

To show the second condition of the induction we fix k and study the following rational function that stems from equation (3.5).

$$L(x) = \frac{1}{(1-x)^{k-1}}\left[1 - \frac{1-x}{(k-1)x}\right].$$

The function $L(x)$ is monotonically increasing in the range $[0, 1]$. To see this note that the derivative of $L(x)$ is

$$L'(x) = \frac{k-1}{(1-x)^k} + \frac{k-2}{(1-x)^{k-1}}\frac{1}{(k-1)x} + \frac{1}{(1-x)^{k-2}}\frac{1}{(k-1)x^2}.$$

Consider the rational functions

$$\begin{aligned} \underline{R}(x) &= L(x) - \frac{1}{(1-\ell_{k-1})^{k-2}}, \\ \overline{R}(x) &= L(x) - \frac{1}{(1-u_{k-1})^{k-2}}. \end{aligned}$$

Both $\underline{R}(x)$ and $\overline{R}(x)$ are monotonically increasing in the range $[0, 1]$. Therefore, if $\underline{R}(\ell_k) < 0$ then

$$\underline{R}(x) \geq 0, x \in [0,1] \implies x > \ell_k.$$

And similarly, if $\overline{R}(u_k) > 0$ then

$$\overline{R}(x) \leq 0, x \in [0,1] \implies x < u_k.$$

We set $\ell_k = \frac{1}{\sqrt{k}+1}$ and $u_k = \frac{2}{\sqrt{k}}$ and show that $\underline{R}(\ell_k) < 0$ and $\overline{R}(u_k) > 0$.

For the induction base we use the transmission probability at q_2. To compute q_2 we use the equation $F_{k-1} = \left(1 - \frac{1-q_k}{(k-1)q_k}\right)\left(\frac{1}{(1-q_k)}\right)^{k-1}$. The expected latency of a single agent is $F_1 = 1$. Therefore, for $k = 2$ this equality translates to $1 = \left(1 - \frac{1-q_2}{q_2}\right)(1-q_2)^{-1}$, which has a unique solution $q_2 = 0.5(\sqrt{5}-1)$ in the region $[0,1]$. For the induction base we notice that $1/(\sqrt{2}+1) < q_2 < 1.5/\sqrt{2}$.

Proposition 3.14. $\underline{R}(\ell_k) < 0$

Proof. By the monotonicity property of $\underline{R}$, $\underline{R}(\ell_k) < \underline{R}(\ell_{k-1})$. Thus, it suffices to show that $\underline{R}(\ell_{k-1}) = \underline{R}(\frac{1}{\sqrt{k-1}+1}) < 0$.

$$\begin{aligned}
R(\frac{1}{\sqrt{k-1}+1}) &= \frac{1}{\left(1-\frac{1}{\sqrt{k-1}+1}\right)^{k-1}}\left[1 - \frac{1-\frac{1}{\sqrt{k-1}+1}}{(k-1)\frac{1}{\sqrt{k-1}+1}}\right] - \frac{1}{(1-\frac{1}{\sqrt{k-1}+1})^{k-2}} \\
&= \frac{1}{\left(1-\frac{1}{\sqrt{k-1}+1}\right)^{k-1}}\left[1 - \frac{1}{\sqrt{k-1}} - \left(1 - \frac{1}{\sqrt{k-1}+1}\right)\right] \\
&= \frac{1}{\left(1-\frac{1}{\sqrt{k-1}+1}\right)^{k-1}}\left[-\frac{1}{\sqrt{k-1}} + \frac{1}{\sqrt{k-1}+1}\right] \\
&< 0.
\end{aligned}$$

∎

For the proof of the second part of Theorem 3.12 we need the following lemmata.

Claim 3.15. *For every integer $x > 0$, $\left(1+\frac{1}{x}\right)^x < e < \left(1+\frac{1}{x}\right)^{x+1}$*

Proof. The left side of the inequality follows directly from the known inequality $1 + y < e^y$ for every real y, by replacing y with $1/x$.

For the inequality on the right side we define the function $h(x) = (1 + 1/x)^{x+1}$. Its first derivative is

$$h'(x) = \frac{1}{x^2}(1+\frac{1}{x})^x(1+x)(-1+x\log(1+\frac{1}{x})).$$

Using the relation $1 + 1/x < e^{\frac{1}{x}}$ once more, we get that $\log(1 + \frac{1}{x}) < \frac{1}{x}$, and therefore $x \log(1 + \frac{1}{x}) < 1$. Consequently, the first derivative of $h(x)$ is negative for $x > 0$, and $h(x)$ is decreasing in this region. Combining this with the known fact that $\lim_{x \leftrightarrow \infty} h(x) = e$, we get that $h(x) > e$ for every $x > 0$. ∎

Claim 3.16. *For every integer $k \geq 1$,*

$$\begin{aligned} \sqrt{k} - \sqrt{k-1} &\leq \frac{1}{\sqrt{k}}(\frac{1}{2} + \frac{1}{k}) && (3.7) \\ \sqrt{k} - \sqrt{k-1} &\leq \frac{1}{2\sqrt{k-1}}. && (3.8) \end{aligned}$$

Proof. It follows from the arithmetic geometric mean theorem that

$$\begin{aligned} \sqrt{k-1} + \frac{1}{\sqrt{k}}(\frac{1}{2} + \frac{1}{k}) &= \frac{1}{2}\left(\sqrt{k-1} + \frac{1}{\sqrt{k}}\right) + \frac{1}{2}\left(\sqrt{k-1} + \frac{2}{k\sqrt{k}})\right) \\ &\geq \sqrt{\left(\sqrt{k-1} + \frac{1}{\sqrt{k}}\right)\left(\sqrt{k-1} + \frac{2}{k\sqrt{k}})\right)} \\ &= \sqrt{k - 1 + \frac{\sqrt{k-1}}{\sqrt{k}} + \frac{2}{k^2} + \frac{2\sqrt{k-1}}{k\sqrt{k}}} \\ &\geq \sqrt{k - 1 + \frac{\sqrt{k-1}}{\sqrt{k}} + \frac{2\sqrt{k-1}}{k\sqrt{k}}} \\ &= \sqrt{k - 1 + \frac{\sqrt{k-1}(2+k)}{\sqrt{k}^3}} \\ &= \sqrt{k - 1 + \sqrt{\frac{(k-1)(2+k)}{k^2}}\sqrt{\frac{2+k}{k}}} \\ &\geq \sqrt{k}, \end{aligned}$$

where the last inequality follows since $(2 + k)(k - 1) \geq k^2$ for every $k > 1$. ∎

Proposition 3.17. $\overline{R}(u_k) > 0$

Proof. The value of $\overline{R}$ at the point u_k is

$$\overline{R}(u_k) = \left(\frac{1}{1 - u_k}\right)^{k-1}\left(1 - \frac{1 - u_k}{(k-1)u_k}\right) - \left(\frac{1}{1 - u_{k-1}}\right)^{k-2}$$

hence we need to show that

$$\left(\frac{1}{1-u_{k-1}}\right)^{k-2} < \left(\frac{1}{1-u_k}\right)^{k-1} \left(1 - \frac{1-u_k}{(k-1)u_k}\right),$$

which, after dividing both sides of this inequality by the right hand side, translates to

$$\left(\frac{1-u_k}{1-u_{k-1}}\right)^{k-1} (1-u_{k-1}) \left(1 - \frac{1-u_k}{(k-1)u_k}\right)^{-1} < 1$$

Assigning $u_k = \frac{c}{\sqrt{k}}$, we get that

$$\left(\frac{1-\frac{c}{\sqrt{k}}}{1-\frac{c}{\sqrt{k-1}}}\right)^{k-1} \left(1 - \frac{c}{\sqrt{k-1}}\right)\left(\frac{ck-c}{ck-\sqrt{k}}\right) < 1 \tag{3.9}$$

We show the above holds for $c = 3/2$, for every $k > 2$. For the case $k \leq 270$ we verify inequality (3.9) manually[8]. For $k > 270$ we give an analytical proof.

Step I: *For every* $1 \leq c < 2$*, and* $k \geq 2$

$$\left(1 - \frac{c}{\sqrt{k-1}}\right)\left(\frac{ck-c}{ck-\sqrt{k}}\right) < 1 - \frac{c-\frac{1}{c}}{\sqrt{k}}.$$

$$\begin{aligned}
\left(1 - \frac{c}{\sqrt{k-1}}\right)\left(\frac{ck-c}{ck-\sqrt{k}}\right) - 1 + \frac{c-\frac{1}{c}}{\sqrt{k}} &= \frac{1+c^2(-2+c(\sqrt{k}-\sqrt{k-1}))}{c\sqrt{k}(c\sqrt{k}-1)} \\
&\overset{(a)}{\leq} \frac{1+c^2(-2+c(\frac{1}{2\sqrt{k-1}}))}{c\sqrt{k}(c\sqrt{k}-1)} \\
&= \frac{1+c^2(-2+c(\frac{1}{2\sqrt{k-1}}))}{c\sqrt{k}(c\sqrt{k}-1)} \\
&\overset{(b)}{<} 0
\end{aligned}$$

The denominator of the last term is positive for every $k > 1$ and $c \geq 1$. When $1 < c < 2$, $\frac{c}{2\sqrt{k-1}} < 2$, for every $k > 1$, and therefore the nominator of the last term is negative. The claim then follows.

Step II: In this step we provide an upper bound on the leftmost term in inequality

[8] A short computer program can verify that the statement holds for $2 < k < 270$.

(3.9):

$$\left(\frac{1-\frac{c}{\sqrt{k}}}{1-\frac{c}{\sqrt{k-1}}}\right)^{k-1} = \left(1+\frac{(\sqrt{k}-\sqrt{k-1})c}{\sqrt{k}(\sqrt{k-1}-c)}\right)^{k-1} < \left(1+\frac{c(\frac{1}{2}+\frac{1}{k})}{k(\sqrt{k-1}-c)}\right)^{k-1} \tag{3.10}$$

where the inequality follows from the bound $\sqrt{k}-\sqrt{k-1} < \frac{1}{2\sqrt{k}} + \frac{1}{k\sqrt{k}}$ (see Claim 3.16).

Step III: *In this step we show that for* $c = 3/2$, *and every* $k > 270$,

$$1+\frac{c(\frac{1}{2}+\frac{1}{k})}{k(\sqrt{k-1}-c)} < 1+\frac{c-\frac{1}{c}}{k\sqrt{k}}. \tag{3.11}$$

When $c = 3/2$, the difference between the left hand side and the right hand side in inequality (3.11) evaluates to

$$\frac{1}{6k^2}\left(-5\sqrt{k}+\frac{9(2+k)}{2\sqrt{k-1}-3}\right).$$

We define $f(k) = -5\sqrt{k}+\frac{9(2+k)}{2\sqrt{k-1}-3}$, and

$$g(k) = -5\sqrt{k-1}+\frac{9(3+(\sqrt{k-1})^2)}{2\sqrt{k-1}-3}.$$

Clearly, $g(k) > f(k)$. It is straightforward to verify that the function $g()$ is decreasing in the region $k > 270$, and that $g(270) < 0$. Therefore, $f(k) < 0$ for every $k > 270$, and thus, the claim follows.

Step VI: *For every* $k > 2$,

$$\left(1+\frac{c-\frac{1}{c}}{k\sqrt{k}}\right)^{k-1} < 1\Bigg/\left(1-\frac{c-\frac{1}{c}}{\sqrt{k}}\right)$$

Consider the left hand side of the above inequality:

$$\left(1+\frac{c-\frac{1}{c}}{k\sqrt{k}}\right)^{k-1} \overset{(a)}{<} e^{\frac{(k-1)(c-\frac{1}{c})}{k\sqrt{k}}} \tag{3.12}$$

$$\overset{(b)}{<} e^{\frac{c-\frac{1}{c}}{\sqrt{k}}} \tag{3.13}$$

$$\overset{(c)}{<} 1+\frac{c-\frac{1}{c}}{\sqrt{k}-(c-\frac{1}{c})} \tag{3.14}$$

$$= 1\bigg/\left(1-\frac{c-\frac{1}{c}}{\sqrt{k}}\right), \tag{3.15}$$

where (a) follows from the inequality $(1+1/x)^x < e$, for $0 < x$. The Inequality (b) holds since $e^{\frac{(c-1/c)}{\sqrt{k}}} > 1$, and (c) follow from Claim 3.15.

The claim follows from combining steps (I) to (IV). ∎

In Figure 3.2 (on the left) we show the transmission probabilities of the time-independent equilibrium, as they are numerically computed. The upper and lower curve correspond to the upper bound $u_k = \frac{3}{2\sqrt{k}}$, and the lower bound $\ell_k = \frac{1}{\sqrt{k+1}}$, respectively. The third line, which lies very close to the actual equilibrium transmission probabilities depicts the curve $\frac{\sqrt{2}}{\sqrt{k+1}}$. Figure 3.2 (on the right) depicts the expected latency in this equilibrium as a function of the number of pending agents.

We end this section by asserting that it follows from the upper bound in Theorem 3.12 that the expected latency of the time independent protocol is at most $\exp(O(\sqrt{n}))$.

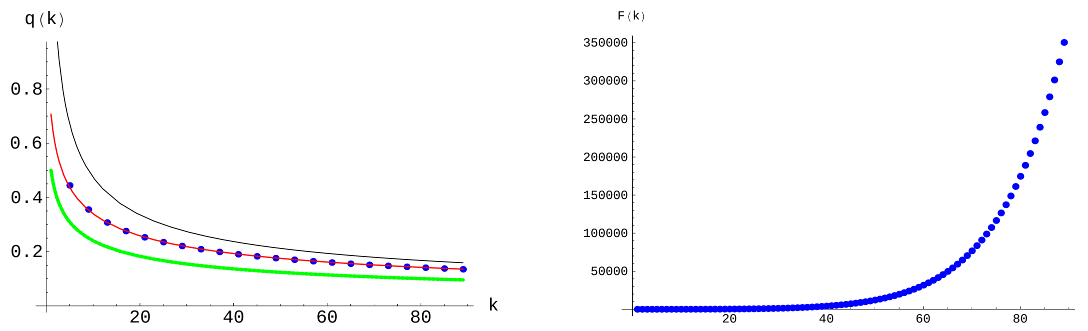
Corollary 3.18. *The expected latency of an agent, when n agents are running the symmetric time independent protocol has an upper bound of* $\exp(O(\sqrt{n}))$, *and a lower bound of* $\exp(\sqrt{n}-1)$

Proof. equation (3.4) constitutes the relation between the expected cost of a time-independent protocol and α_k (F_k has a geometric distribution with parameter α_k. The expected cost of an agent when all agents are using the time-independent protocol is

$$C_{n,0} = F_n = 1/\alpha_n = 1/(1-q_n)^{n-1}, \tag{3.16}$$

which is increasing in q_n in the interval $q_n \in [0,1]$.

Theorem 3.12 yields a lower bound and an upper bound on q_n, for every $n \geq 2$. From the upper bound on q_n, $q_n < 1.5/\sqrt{k}$ we gain an upper bound on the expected latency,



(a) Transmission Probability (b) Expected Cost

Figure 3.2: (a) The transmission probabilities in equilibrium as a function of the number of agents, of the unique non-blocking, time independent equilibrium, for agents with latency cost (described by the blue dots). The transmission probability in equilibrium lies between the curve $u_k = 1.5/(\sqrt{k})$, and the curve $\ell_k = 1/(\sqrt{k}+1)$. The middle curve depicts $\sqrt{2}/(\sqrt{k}+1)$, and is seemingly very close to the actual transmission probabilities of this equilibrium. (b) The expected latency in equilibrium as a function of the number of pending agents.

$$C_{n,0} \leq \left(\left(1-\frac{1.5}{\sqrt{n}}\right)^{n-1}\right)^{-1} = \left(\left(1-\frac{1.5}{\sqrt{n}}\right)^{\frac{\sqrt{n}}{1.5}}\right)^{-(1.5\sqrt{n}-o(1))} = O(\exp(1.5\sqrt{n})).$$

From the lower bound on q_n, $1/(\sqrt{n}+1) < q_n$ we derive a lower bound on the expected latency:

$$C_{n,0} \geq \left(\left(1-\frac{1}{\sqrt{n}+1}\right)^{n-1}\right)^{-1} = \left(\left(1-\frac{1}{\sqrt{n}+1}\right)^{\sqrt{n}+1}\right)^{-(\sqrt{n}-1)} > \exp(\sqrt{n}-1),$$

where the last inequality follows from Claim 3.15. ∎

3.4 Efficient Protocols in Equilibria

Given n agents at time zero, Theorem 3.12 implies that the expected latency for (the unique) symmetric, time-independent, protocol in equilibria is exponential in n. Ergo, the probability that even one agent will be successful within any polynomial time bound is exponentially small.

In contrast, efficient protocols ensure that all n agents succeed in linear time except with exponentially small probability. In this section we give a protocol for contention resolution, which is simultaneously efficient, symmetric, and in equilibrium. Obviously, such a protocol *cannot* be time-independent.

To achieve efficient protocols for agents with latency costs, we turn aside from latency cost protocols to address strategic behavior under deadline cost functions.

We consider two related deadline cost functions, the first is a pure deadline, the 2nd is a combination of latency costs plus a deadline.

$$\Psi_D(t) = \begin{cases} 0 & \text{For } 0 \le t < D; \\ 1 & \text{Otherwise.} \end{cases}$$

Or,

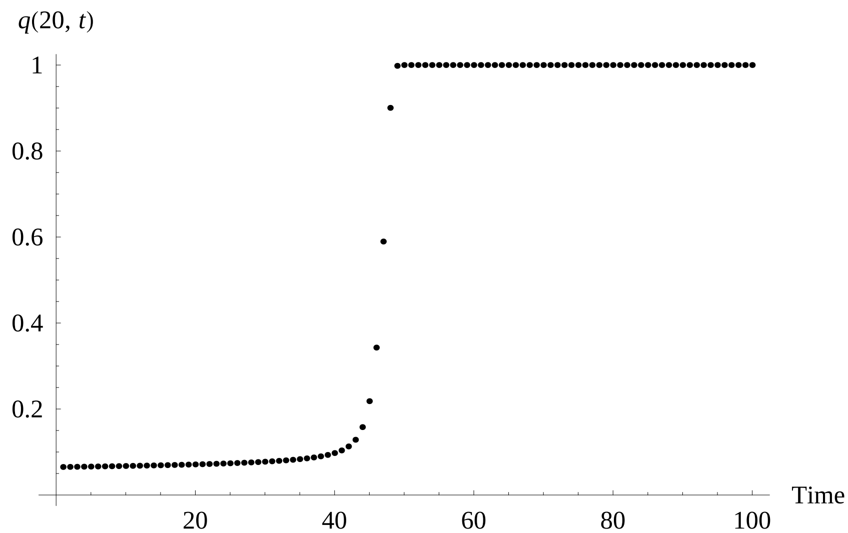
$$\Psi^*_{D,M}(t) = \begin{cases} t & \text{For } 0 \le t < D; \\ M + t & \text{Otherwise.} \end{cases}$$

Time slot D is referred to as the *deadline*.

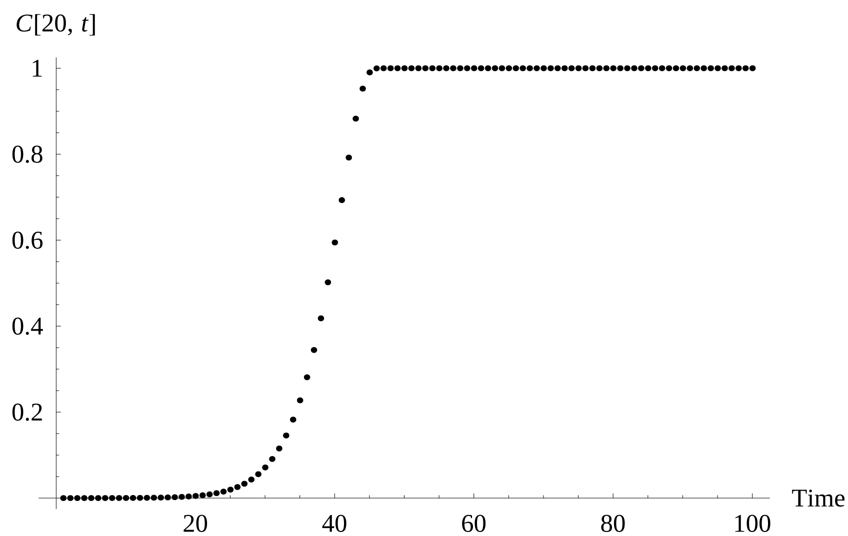
Figure 3.3a shows the numerical solution of the symmetric protocol in equilibrium, for the deadline cost function $\Psi_D(\cdot)$, where the number of agents is $n = 20$, and a deadline is set at time $D = 100$. For a Ψ_D cost function, the equilibrium equations can be solved, as the recursion base is known — at time D, the transmission probability is 1 for any number of agents $1 \le k \le n$. The solution is a matrix $\{q_{k,t}\}_{1\le k\le n \times 1\le t\le D}$, that gives a transmission probability $q_{k,t}$ for any number of pending agents $1 \le k \le n$, and time $t \le D$. To illustrate it graphically we fix the number of agents at 20, and present the transmission probability of 20 pending agents as a function of the time $t \in \{1, \ldots, D\}$. Figure 3.3b shows the expected cost of a pending agent when overall 20 agents are pending, as a function of the time (which equals the probability of a successful transmission prior to the deadline).

There seems to be a phase transition of the transmission probabilities approximately at time $t^* = D - 3.5n$. Prior to t^* the transmission probability is very close to 1/20; After t^*, the transmission probability rapidly increases to 1. Likewise, in 3.3b, prior to t^*, the probability of failing to transmit before the deadline is almost 0; after t^* the failure probability arises to 1. Similar phenomena appear for other values of n.

Unfortunately, we are unable to prove this empirical observation. What we can show



(a) Equilibrium Transmission Probability



(b) Probability of Failing to Transmit Prior to the Deadline

Figure 3.3: This figure presents numerical solutions of the recursive equilibrium equations for a game where agents have the deadline cost function; the deadline is set at time $D = 100$. In (a) the transmission probability in equilibrium is demonstrated, for a fixed number of pending agents $k = 20$ (*i.e.*, the x-axis represents time, and the y-axis represents transmission probability in equilibrium when 20 agents are pending). In (b) the y-axis represents the expected cost for a pending agent at time t, when overall 20 agents are pending. Notice the threshold phenomenon in both Figures 3.3a and 3.3b — the transition between transmission probability $1/k$ and 1 in (a), and accordingly the transition from success with high probability to an almost sure failure, is very rapid.

is that for time slots close to the deadline (say $D-(1+\epsilon)n$), the transmission probability is arbitrarily close to one. We also show that the probability that even a single agent is delayed until the deadline is negligible if the deadline is at least $15n$ time slots away. This implies that agents transmit with probability $O(1/k)$ during a large fraction of the time slots $t \in \{D-15n, \ldots, D\}$.

The main result of this section is the following theorem that says that for deadlines at least $15n$ time slots away, there exist protocols in equilibria such that all agents succeed before the deadline with high probability:

Theorem 3.19. *For every time D, and number of agents $D > n$ there exists a symmetric protocol Q_D in equilibrium for the Ψ_D cost function, such that if the deadline $D > 15n$ then the probability that not all agents succeed prior to the deadline is negligible ($1/e^{\Theta(D)}$).*

We begin with the simple case of 2 agents, Alice and Bob. After the deadline expires, all strategies are in equilibrium as they have no effect on the cost.

Assume a symmetric protocol in equilibrium, and consider the transmission probability $q_{2,D-1}$ used by both of the 2 pending agents at time $D-1$. We show that $q_{2,D-1}=1$. Corollary 3.10 implies that $q_{2,D-1} \geq 1/2 > 0$. Assume that $q_{2,D-1} < 1$ — this says that both transmitting and remaining quiescent are in the support of this mixed strategy in equilibrium.

Consider the pure strategy in which Alice chooses to transmit deterministically at time $D-1$. The expected cost to Alice is then equal to the probability that Bob also chooses to transmit, $q_{2,D-1} < 1$. If Alice chooses the pure strategy of remaining quiescent at time $D-1$ then she is doomed to reach the deadline and her cost is exactly 1. *I.e.*, we have unequal expected costs for two pure strategies in the support, contradicting that $q_{2,D-1} < 1$.

It follows that for every symmetric protocol in equilibrium $q_{2,D-1}=1$. This can be further generalized to whenever k agents are pending at one of the last $k-1$ time slots prior to the deadline, $q_{k,D-k+1} = q_{k,D-k+2} = \cdots = q_{k,D-1} = 1$. One can prove that

Lemma 3.20. *Consider n agents with deadline cost function Ψ_D. Let $\langle q\rangle^n$, be a symmetric protocol in equilibrium for such agents, then, for all $0 \leq t \leq D$ and for any $k > D-t$ we have $q_{k,t}=1$.*

Likewise, there exists a symmetric protocol in equilibrium for such agents, $\langle q\rangle^n$, such that for all $0 \leq t < D$ for every $k \leq D-t$ we have $q_{k,t} < 1$.

Proof. The proof is by double induction, the outer induction is on the number of agents, and the inner induction is on the number of time slots remaining until the deadline.

We've previously argued that $q_{2,D-1} = 1$ but the same argument also shows that $q_{k,D-1} = 1$ for all $k \leq n$.

The induction hypothesis for the outer induction is that for some $k < n$, and all $t > D-k$, $q_{k,t} = 1$. The inductive step is to prove that for all $t > D-(k+1)$, $q_{k+1,t} = 1$. The base case for the outer induction is that $q_{2,D-1} = 1$.

To prove the outer induction for $k+1$, we use an inner induction on the number of time slots remaining until the deadline. The base for the inner induction is that $q_{k+1,D-1} = 1$. The inner induction hypothesis is that $q_{k+1,t} = 1$ for some $t > D-(k+1)+1 = D-k$. Let Alice be one of these $k+1$ pending agents that plays Quiescent at time $t-1$. Even if some agent other than Alice is successful at time $t-1$, there will still be $\geq k$ pending agents (including Alice) at time t. By the outer induction, Alice is doomed not to succeed before the deadline, and thus $q_{k+1,t-1} < 1$ cannot be in equilibrium. ∎

Lemma 3.20 implies that there is some probability $p > 0$ that all k pending agents at time $D-k$ will succeed before the deadline. We remark that given k pending agents at time $D-k$, the probability of even one agent being successful before the deadline D is negligible (super-exponentially small in k). In comparison, Theorem 3.19 says that given k pending agents at time $D-15k$, then all k agents will succeed before D, except with negligible probability.

It is natural to consider the case of $k = 2$ pending agents:

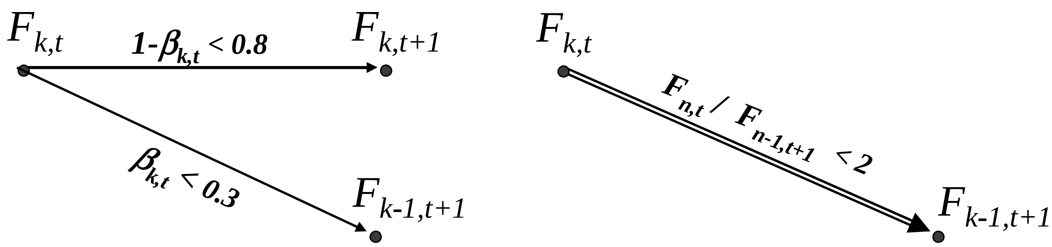
Lemma 3.21. *Symmetric protocols in equilibrium for the Ψ_D deadline cost function have*

$$q_{2,t} = \begin{cases} 1/2 & \text{for } 0 \leq t \leq D-2, \\ 1 & \text{otherwise.} \end{cases}$$

Also, the expected cost $C_{2,t} = (1/2)^{D-t-1}$ for $t \leq D-1$.

Proof. For the deadline cost function Ψ_D the expected future cost of a single agent at time $t \leq D-1$, $F_{1,t} = 0$. However, the expected future cost for one of two pending agents can never be 0 (Since $\langle q \rangle^2$ is a symmetric protocol). It follows from Lemma 3.8 that for $t \leq D-2$ we have

$$q_{2,t} = \frac{1}{k-(k-1)\frac{F_{k-1,t+1}}{F_{k,t+1}}} = \frac{1}{2-\frac{0}{F_{2,t+1}}} = 1/2.$$



(a) Good Edges (b) A Doubling Edge

Figure 3.4: (a) Edges of type "Good" with weight strictly less than 1. (b) An edge of type "Doubling"; the weight at most 2.

For deadline cost function Ψ_D, the expected cost $C_{2,t}$ equals the probability of remaining unsuccessful until the deadline, $C_{2,t} = (1/2)^{D-t-1}$. ∎

For any deadline D and number of pending agents k we can give a recursive description of the probabilities in equilibrium, $q_{k,t}$, this gives an algorithm for the computation of such $q_{k,t}$, but we now turn to the asymptotic analysis of such equilibria. Obviously, for all $0 \le t < D$, $2 \le k \le n$, either $F_{k,t} \le 2F_{k-1,t}$ or $F_{k,t} > 2F_{k-1,t}$. In the latter case, it follows from Lemma 3.8 that

$$q_{k,t-1} = \frac{1}{k-(k-1)\frac{F_{k-1,t}}{F_{k,t}}} < \frac{1}{k-(k-1)/2} < 2/k.$$

We now describe a rooted tree, $T = (V, E)$, with weights on the edges. For edge z, $w(z)$ is the weight of the edge (as illustrated in Figure 3.5). Vertices $v \in V$ have labels $\ell(v) = (k, t)$ for some $1 \le k \le n$, $0 \le t \le D$. Not all the $n(D+1)$ possible labels need appear on some vertex $v \in V$, and the same label may appear multiple times ($\ell(v) = \ell(v')$, $v \ne v'$). The root vertex r is assigned the label $\ell(r) = (n, 0)$, and is the only vertex so labeled.

Given $v \in V$, with $\ell(v) = (k, t)$, $2 \le k \le n$, $0 \le t < D$, we attach descendants to v as follows (as illustrated in Figure 3.4):

- If $F_{k,t+1} \le 2F_{k-1,t+1}$, then v has one descendant, x, with $\ell(x) = (k-1, t+1)$. Edge (v, x) is given weight $w(v, x) = F_{k,t}/F_{k-1,t+1}$. Note that $w(v, x) \le 2$, since $F_{k,t} \le F_{k,t+1} \le 2F_{k-1,t+1}$, where the first inequality follows Lemma 3.7. Such edges, where v has a single descendant, are called *doubling edges* and the set of all such edges is denoted by E_d.

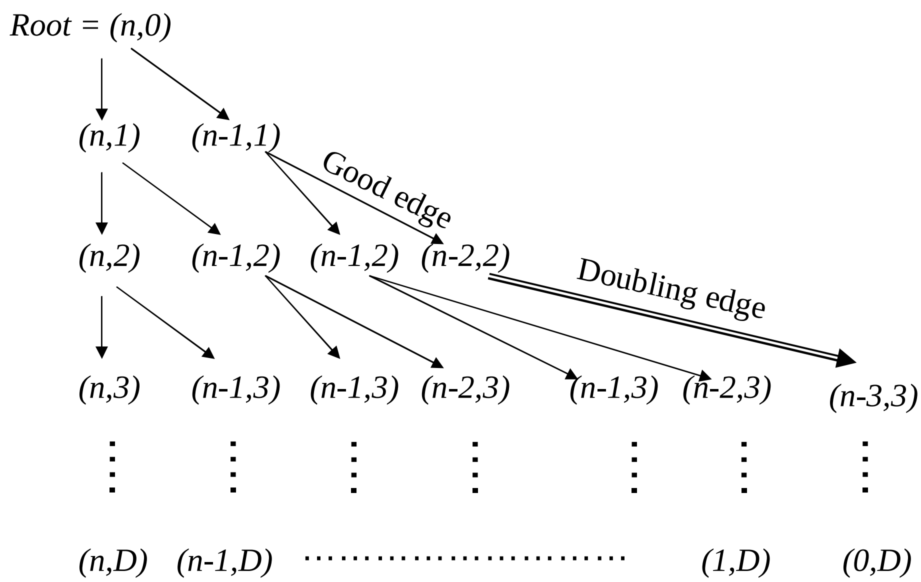


Figure 3.5: An example of a tree formed as described in the analysis of Theorem 3.19. Notice that multiple vertices may share the same label.

- Otherwise, v has two descendants, y, and x, where $\ell(y) = (k-1, t+1)$ and $\ell(x) = (k, t+1)$. The weight $w(v, y) = \beta_{k,t}$ (See equation 3.2 to recall the definition of β_{kt}), and $w(v, x) = 1 - \beta_{k,t}$.

Notice that an edge e either connects a vertex labeled (k, t) with a vertex labeled $(k-1, t+1)$, or with a vertex $(k, t+1)$.

Let L_0 be the set of vertices $v \in V$ with labels $\ell(v) = (1, t)$, $0 \leq t < D$. Let L_1 be the set of vertices $v \in V$ with labels $\ell(v) = (k, D)$, $1 \leq k \leq n$. The set $L_0 \cup L_1$ is exactly the set of leaves in T.

For any leaf v, where $\ell(v) = (k, t)$, we define the real value $c(v) = C_{k,t}$. *I.e.*, for $v \in L_0$, $c(v) = 0$ and for $v \in L_1$, $c(v) = 1$. An internal vertex v with two descendants, x, y, has $c(v) = w(v, x)c(x) + w(v, y)c(y)$, an internal vertex v with one descendant, x, has $c(v) = w(v, x)c(x)$. It follows from the recursive construction of T and from the recursive evaluation of the vertices cost c that $c(r) = C_{n,0}$.

For a leaf v in T let $P(v)$ denote the set of edges along the path from the root r to v.

One can rearrange the recursive summation for the value of $c(r)$ as follows:

$$\begin{aligned} c(r) &= \sum_{v \in L_0} \left(c(v) \prod_{g \in P(v)} w(g) \right) + \sum_{v \in L_1} \left(c(v) \prod_{g \in P(v)} w(g) \right) \\ &= \sum_{v \in L_1} \left(\prod_{g \in P'(v)} w(g) \prod_{g \in P''(v)} w(g) \right), \end{aligned}$$

where $P'(v) = P(v) \cap E_d$ and $P''(v) = P(v) \cap (E - E_d) = P(v) - P'(v)$.

Next, we set an upper bound on the weight of an edge $g \in E - E_d$, of β defined as $\beta = \max\{\beta_{k,t}, 1 - \beta_{k,t}\} \leq 1 - \frac{2}{e^2}$, which is strictly less than 1.

Lemma 3.22. *For all edges $g \in E - E_d$, $w(g) \leq \beta$.*

Proof. A vertex labeled (k,t) that has two descendants implies that $q_{k,t} < \frac{2}{k}$. The term $\beta_{k,t} = (k-1)q_{k,t}(1-q_{k,t})^{k-2}$ as a function of $q_{k,t}$ is monotonically decreasing in the range $[\frac{1}{k-1}, 1]$. Hence, for $k > 2$,

$$\frac{2}{e^2} < (k-1)\frac{2}{k}\left(1 - \frac{2}{k}\right)^{k-2} < \beta_{k,t} < (k-1)\frac{1}{k-1}\left(1 - \frac{1}{k-1}\right)^{k-2} \leq \frac{1}{2}.$$

∎

The size of L_1 is no more than $\sum_{k=1}^{n} \binom{D}{k}$ which is less than $n\binom{D}{n}$ for a deadline $D \geq 2n$. The product of edges weights for edges in E_d is at most 2^n, since there are at most n such edges (this follows since a doubling edge decreases the first coordinate of the label by one, and vertices with labels $(1,t)$ are leaves). The product of edge weights for edges in $E - E_d$ along some path from r to $v \in L_1$ decreases exponentially with the path length, which is at least $D - n$.

It follows that

$$\prod_{g \in P(v) \cap (E - E_d)} w(g) \leq \beta^{D-n}, \text{ and thus}$$

$$c(r) \leq n\binom{D}{n} 2^n \beta^{D-n}.$$

For $D = bn$, $\binom{D}{n} < (eb)^n$ and therefore $c(r) \leq e^{\Theta(n \ln b) - nb\ln(1/\beta)}$ which is exponentially decreasing in $D = bn$. The value $c(r)$ is the probability that a specific agent will fail to successfully transmit before the deadline. It follows that the probability that all n agents are successful prior to the deadline is at least $1 - nC_{n,0}$. Setting $b = 15$ suffices to have the cost $c(r)$ diminishes in n when the deadline is set to bn.

Remark 3.23. *The choice of the constant 2 in the analysis (i.e., vertex (k,t) has a single descendant if $F_{k,t+1} \leq 2F_{k-1,t+1}$ and two descendants otherwise) was arbitrary. Optimizing, we can reduce the requirement on the deadline in Theorem 3.19 from being at least $15n$ to being at least $12.3n$. This is done by changing the construction rule so that a vertex (k,t) has a single descendant if $F_{k,t+1} < 3.4F_{k-1,t+1}$ and two descendants otherwise.*

An Efficient Protocol for Agents with Linear Latency Cost

We now describe an efficient protocol $Q^*_{D,M}$, in equilibrium for latency cost agents. Our algorithm "tricks" the agents into acting as if they have deadline costs $\Psi^*_{D=100n,M=\exp(9n)}$. This is achieved as follows: any $k \geq 3$ pending agents at time $t = D$ transmit continuously for M time slots, following which they revert to the time independent protocol for latency cost (see Section 3.3, and the discussion at the end of this Section). Then, Theorem 3.19 applies and we can conclude that the protocol is not only in equilibrium but also efficient.

Effectively, this means that the utility function for the agents is no longer latency costs but rather $\Psi^*_{D,M}$, where M is chosen to be very large, $M = \exp(O(n))$.

Consequently, the future cost of this protocol for $k \geq 3$ agents at time D, $F_{k,D}$, is at least M. Precisely, it is M plus the expected delay of the time independent protocol derived in Section 3.3, which is $O(\exp(1.5\sqrt{n}))$ (from Corollary 3.18). Define vertices v labeled (k, D), $k \geq 3$, to be leaves of a tree with $c(v) \geq M$ (analogously to the set L_1 above). Define vertices v labeled $(1, j)$, $j \leq D$, to be leaves with $c(v) = j$. Likewise, vertices $(1, D)$ and $(2, D)$ are leaves with $c(v) = O(1)$. Define the weights of the edges as above, by construction, $C_{n,0} = c(r)$.

Theorem 3.24. *Protocol $Q^*_{D,M}$ is in equilibrium for the $\Psi^*_{D,M}$ cost function. If the deadline $D > 100n$ and $M = \exp(9n)$ then the probability that not all agents succeed prior to the deadline is negligible ($1/e^{\Theta(D)}$).*

The analysis of this protocol is closely related with the analysis of the protocol for the pure deadline cost function Ψ_D — Intuitively, as M increases we expect $\Psi^*_{D,M}$ to become more and more "similar" to Ψ_D. Specifically, we set $M = e^{9n}$. For this choice of M we could still choose the factor b large enough, so that the second term in the sum of $c(r)$ (equation (3.17)), diminishes with n.

Unlike the analysis for deadline cost function, where agents successful prior to the deadline had a cost 0, the first term of the left hand side in equation (3.17), does not nullify. Nevertheless, we could still bound it from above by the number of distinct paths that lead to a vertex in L_0, times the value of such vertex, which cannot exceed D, times the weight of the path, which cannot exceed 2^n.

We get that for large enough b, $c(r) << M$, from which it follows that the probability that an agent remains in the system at time D is negligible.

It is important to remark that the expected cost is not linear, rather - the protocol completes in linear time with all but negligible probability.

Formally, let us set $b = 100$, and $M = e^{9n}$. The cost of $C_{n,0} = c(r)$ consists of the cost contributed by L_0 leaves (with cost at most D), and that contributed by the L_1 leaves (with cost at least M). The contribution by the L_0 leaves has an upper bound of e^{8n}:

$$\begin{aligned}
\sum_{v\in L_0}\left(c(v)\prod_{g\in P(v)} w(g)\right) &\leq \sum_{v\in L_0} 2^n D \\
&\leq \sum_{m=n}^{D}\binom{m}{n}2^n D \\
&\leq D\binom{D}{n}2^n D \\
&\leq D^2(be)^n 2^n \\
&\leq (be)^{3n} \\
&= e^{n\log b+3} \\
&\leq e^{8n}
\end{aligned}$$

The cost of a leaf in L_1 is at most M plus the expected latency of the time independent protocol for the linear cost, which is at most 2^n. The contribution from the L_1 leaves diminishes with n, for our choice of b:

$$\begin{aligned}
\sum_{v\in L_1}\left(c(v)\prod_{g\in P(v)} w(g)\right) &< n(eb)^n 2^n \beta^{-n}\beta^{bn}(M+\exp(1.5\sqrt{n}))) \\
&\leq e^{\log(b)(n+11)}e^{\log(\beta)bn} \\
&\leq e^{-n}
\end{aligned}$$

Discount Factor	**Transmission Probability**	**Efficiency**
$0 < \delta < 1$	$1 - \delta \geq q_k$	$\exp(O(n))$
$\delta = 1$	$q_k = \Theta(1/\sqrt{k})$	$\exp(O(\sqrt{n}))$
$1 < \delta < 2$, $k = 2$	$q_2 = \frac{-1+\sqrt{1+4\delta}}{2\delta}$	$\Theta(1)$
$1 < \delta < 2$, $k > 2$	unknown	unknown — if a protocol exists, it attains linear expected latency
$2 \leq \delta$	undetermined — the expected cost of any protocol is ∞	undetermined

Table 3.2: This table summarizes our results regarding time independent symmetric equilibrium for cost functions of the form $\Psi(t) = \sum_{\tau=0}^{t-1} \delta^t$. The second column tells us the transmission probability when there are k agents pending, and the third column is the expected latency of the protocol. For $\delta > 2$, and any number of agents $k \geq 2$, any protocol yields an infinite expected cost. This does not fit our cost model, where an agent's preference order between protocols is based on their expected cost. For $1 < \delta < 2$ we can show the existence of a protocol in equilibrium only for two agents.

In conclusion, we get that the expected cost for an agent in of the $\Psi^*_{D,M}$ protocol is $O(e^{8n})$, which is significantly less than the expected cost for agents that are not successful prior to the deadline (in which case the cost is greater than M). Thus, due to Markov inequality, an agent is successful prior to D, with probability of at least $c(r)/M < e^{-n}$.

3.5 Discounted Latency Cost

In Section 3.3 we analyzed the performance of the unique symmetric, time-independent protocol in equilibrium for the linear latency cost function. In this section we show that such a symmetric equilibrium in time-independent strategies exists for a broader class of cost functions of the form $\Psi(t) = \sum_{j=0}^{t} \delta^j$, where $\delta < 1$ is a discount factor. Discounting is common in the analysis of repeated games, and have the interpretation that the extra cost of delaying another time slot diminishes over time.

For such cost functions it is convenient to consider a *rent* function $\psi(t)$ for the marginal cost at time slot t, *i.e.*, $\psi(t) = \delta^t$.

It turns out that $\delta = 1$ is a critical point for the expected time it takes to resolve contention in symmetric equilibrium. When $\delta < 1$ the transmission probability in equilibrium is bounded from below by a constant that depends only on δ and therefore the expected time for transmission is exponential in the number of agents. The case $\delta = 1$ has already been treated in Theorem 3.12, where we show that time independent protocols for linear costs (*i.e.*, $\delta = 1$) have expected latency $\Theta(\exp(\sqrt{n}))$.

When $\delta < 1$ the cost of an agent is bounded from above by a constant $1/(1-\delta)$, no matter what protocol is in use, and even if her delay is infinite[9]. However, the expected time until a successful transmission is exponential in the number of agents, due to a very aggressive transmission rate. Intuitively, agents with decaying costs are aggressive in every step as the rent of this step dominates the rest of the expected payments.

Theorem 3.25 characterizes the transmission probability in equilibrium for a whole range of $0 < \delta < 1$.

Theorem 3.25. *There exits a unique time-independent, symmetric, and non-blocking protocol $\langle q\rangle^n$ in equilibrium $q = \langle q_1, \ldots, q_n\rangle$ for rent functions of the form δ^t, for $0 < \delta < 1$. Furthermore, the transmission probability $q_k < 1 - \delta,\ 2 \le k \le n$.*

For the proof of Theorem 3.25 we use the fact that time is irrelevant in the description of equilibrium in time-independent strategies, for such cost functions, as shown in Lemma 3.26.

Lemma 3.26. *For a symmetric time-independent strategy $q = \langle q_1, \ldots, q_n\rangle$, and a rent function $\psi(t) = \delta^t$,*

$$C_{k,t+1} = 1 + \delta C_{k,t}\ ;\ F_{k,t+1} = \delta F_{k,t}$$

Proof. Consider a rent function δ^t, $\delta < 1$. First, observe that for an exponential rent function,

$$\Psi(t+1) = \sum_{\tau=0}^{t} \delta^\tau = 1 + \sum_{\tau=1}^{t} \delta^\tau = 1 + \delta\Psi(t)\ .$$

Also, at every slot, the expected time until successful transmission T, for some pending agent Alice, depends only on the number of pending agents. Recall that the random variable $T_{k,t}$ describes the time of a successful transmission when there are k agents pending at time t. For time-independent strategies, the probability of a successful transmission in i time slots depends only the number of pending agents, *i.e.*,

$$\Pr[T_{k,t} = t + i] = \Pr[T_{k,s} = s + i]\ .$$

[9]Simply because the sum of the geometric series $1 + \delta + \delta^2 \ldots < 1/(1-\delta)$.

This means that for every i,k,t, the random variable $T_{k,t+i}$ has the same distribution as the random variable $(T_{k,t} + i)$,

$$\Pr[(T_{k,t}+i) = t+M] = \Pr[T_{k,t} = t+M-i] = \Pr[T_{k,t+i} = t+i+M-i] = \Pr[T_{k,t+i} = t+M] \ ,$$

and therefore,

$$C_{k,t+1} = \mathrm{E}[\Psi(T_{k,t+1})] = \mathrm{E}[\Psi(T_{k,t} + 1)] = \mathrm{E}[1 + \delta\Psi(T_{k,t})] = 1 + \delta\mathrm{E}[\Psi(T_{k,t})] = 1 + \delta C_{k,t} \ .$$

The future expected cost in slot $t + 1$ with k pending agents can be written in terms of the future expected cost in slot t with k pending agents,

$$F_{k,t+1} = C_{k,t+1} - \Psi(t + 1) = 1 + \delta C_{k,t} - (1 + \delta\Psi(t)) = \delta F_{k,t} \ .$$

∎

Proof of Theorem 3.25. For the existence and uniqueness part of Theorem 3.25 we take a similar approach to the one we used in the proof of Theorem 3.12. To compute $F_{k,t}$ when $k \geq 2$ we consider agent Alice, one of $k \geq 2$ pending agents at time t. Assume Alice deviates from q and continuously transmits until successful. This pure strategy is in the support of q, therefore it has an expected cost equal to that of q. As Alice continuously transmits, no agent other than Alice can succeed while Alice is pending. Recall that $\alpha_k = (1 - q_k)^{k-1}$ is the probability Alice is successful in a slot [10].

Alice's latency is geometrically distributed with parameter α_k. Her actual cost is not

[10] the index t is dropped from $\alpha_{k,t}$, since the transmission probability is time independent.

geometrically distributed, but can nevertheless has an explicit form:

$$\begin{aligned}
F_{k,t} &= \left(\sum_{\tau=1}^{\infty}(1-\alpha_k)^{\tau-1}\alpha_k\Psi(\tau+t)\right)-\Psi(t) &(3.17)\\
&= \sum_{\tau=1}^{\infty}(1-\alpha_k)^{\tau-1}\alpha_k\left(\Psi(\tau+t)-\Psi(t)\right) &(3.18)\\
&= \sum_{\tau=1}^{\infty}(1-\alpha_k)^{\tau-1}\alpha_k\left(\sum_{j=0}^{\tau+t-1}\delta^j-\sum_{j=0}^{t-1}\delta^j\right) &(3.19)\\
&= \sum_{\tau=1}^{\infty}(1-\alpha_k)^{\tau-1}\alpha_k\delta^t\sum_{j=0}^{\tau-1}\delta^j &(3.20)\\
&= \alpha_k\delta^t\sum_{\tau=1}^{\infty}(1-\alpha_k)^{\tau-1}\frac{\delta^\tau-1}{\delta-1} &(3.21)\\
&= \frac{\alpha_k\delta^t}{\delta-1}\left(\sum_{\tau=1}^{\infty}(1-\alpha_k)^{\tau-1}\delta^\tau-\sum_{\tau=1}^{\infty}(1-\alpha_k)^{\tau-1}\right) &(3.22)\\
&= \frac{\alpha_k\delta^t}{\delta-1}\left(\delta\frac{1}{1-\delta(1-\alpha_k)}-\frac{1}{1-(1-\alpha_k)}\right) &(3.23)\\
&= \delta^t\frac{1}{1+\delta(\alpha_k-1)} &(3.24)
\end{aligned}$$

Notice that in passing from Line 3.22 to Line 3.23 we used the fact that both $0<\delta<1$, and $0<1-\alpha_k<1$, to assert convergence of the corresponding infinite summations.

Combining equations (3.17-3.24), and equation (3.5) yields $F_{k,t}$ as a function of q_k:

$$F_{k-1,t} = \delta^t\frac{1}{1+\delta((1-q_k)^{k-1}-1)}(1-\frac{1-q_k}{(k-1)q_k}) \quad (3.25)$$

We show now by induction on the number of pending agents that for every $0<\delta<1$, a non-blocking time-independent protocol in equilibrium exists. For the induction base we will show that $0<q_2<1$. Recall that in equilibrium the transmission probability of a single pending agent, at any time t, is 1. Accordingly, her future cost is $F_{1,t}=\Psi(t+1)-\Psi(t)=\delta^t$. Together with the last equality, we have the following equation when $k=2$, for every $t\geq 0$:

$$\delta^t = \delta^t\frac{1}{1-\delta q_2}(1-\frac{1-q_2}{q_2}). \quad (3.26)$$

Equation (3.26) has a unique positive solution for q_2, for every $0<\delta<1$, $q_2=\frac{-1+\sqrt{1+4\delta}}{2\delta}$.

Thus, $0 < q_2 < 1$, *i.e.*, strictly mixed transmission probability.

For the induction step assume that $F_{k-1,t}$ is strictly smaller than $1/(1-\delta)$. Consider the right hand side of equation (3.25), and let f denote the function

$$f(q_k) = \delta^t \frac{1}{1+\delta((1-q_k)^{k-1}-1)}(1-\frac{1-q_k}{(k-1)q_k}),$$

which is continuous and increasing. For every $2 \leq k, 0 < \delta < 1$ we have $f(0) = 0$, and $\lim_{q_k \to 1} f(q_k) = 1/(1-\delta)$. Namely, $q_k = f^{-1}(F_{k-1,0})$ is well defined when $F_{k-1,t}$ belongs to the half open interval $[0, \frac{1}{1-\delta})$.

We conclude that a symmetric, non-blocking, and time independent protocol exists, for every $0 < \delta < 1$, and following the fact f^{-1} is well defined, it is also unique.

We now turn to show the lower bound on the transmission probabilities. We first bound $F_{k,t}$, the expected additional cost at state (k,t), from above.

$$F_{k,t} < \sum_{\tau=t+1}^{\infty} \delta^{\tau-1} = \frac{\delta^t}{1-\delta} \; . \tag{3.27}$$

Now, let us bound $F_{k,t}$ from below. Consider a miraculous scenario, where there is a successful broadcast on each of the next k slots. Let OPT denote the average future cost on this scenario. Clearly, the true $F_{k,t}$, cannot be less than OPT.

$$
\begin{aligned}
F_{k,t} \geq \mathrm{OPT} &= \frac{1}{k}\sum_{i=0}^{k-1}\Psi(i+(t+1))-\Psi(t) \\
&= \frac{1}{k}\sum_{i=0}^{k-1}\sum_{j=t}^{i+t}\delta^j \\
&= \frac{1}{k}\sum_{i=0}^{k-1}\delta^t\sum_{j=0}^{i}\delta^j \\
&= \frac{\delta^t}{k}\sum_{i=0}^{k-1}\frac{1-\delta^{i+1}}{1-\delta} \\
&= \frac{\delta^t}{k}\cdot\frac{1}{1-\delta}\sum_{i=0}^{k-1}1-\delta^{i+1} \\
&= \frac{\delta^t}{1-\delta}-\frac{\delta^{t+1}}{k(1-\delta)}\sum_{i=0}^{k-1}\delta^i \\
&= \frac{\delta^t}{1-\delta}-\frac{\delta^{t+1}}{k(1-\delta)}\cdot\frac{1-\delta^k}{1-\delta} \\
&= \frac{\delta^t}{1-\delta}(1-\frac{1}{k}\delta(\frac{1-\delta^k}{1-\delta}))
\end{aligned}
$$

The last inequality combined with inequality (3.27), implies that, $\frac{F_{k-1,t}}{F_{k,t}} > 1-\frac{1}{k-1}\delta(\frac{1-\delta^{k-1}}{1-\delta})$. Using Lemma 3.8, we get that

$$
q_k = \frac{1}{k-(k-1)\frac{F_{k-1,t}}{F_{k,t}}} > \frac{1}{k-(k-1)(1-\frac{1}{k-1}\delta(\frac{1-\delta^{k-1}}{1-\delta}))} > \frac{1}{1+\frac{\delta}{1-\delta}} = 1-\delta \ ,
$$

which settles the upper bound on q_k for the case $0<\delta<1$. ∎

Impatient Agents. A discounted factor, $0<\delta<1$ means that an agent values one unit of time now, more than he would in the next period. If we choose $\delta>1$, then the cost function would describe an *impatient* agent — the cost of waiting increases exponentially with every step. How would such a cost function affect the equilibrium?

In the proof of existence and uniqueness of the symmetric, time-independent, non-blocking equilibrium above, we did not explicitly use the fact that $\delta<1$. However, it has been implicitly when we treated $F_{k,t}$ as finite. On the contrary, when $\delta > 1$, the expected cost need not be finite. In fact, for large enough δ, no protocol, in equilibrium or not, yields a finite expected cost.

To show this, reconsider equation (3.17), which describes the expected cost of a protocol with exponential rent function $\psi(t) = \delta^t$. Notice that nothing changes in the developing of equation (3.17) until line (3.24), except that the summation in line (3.17) need not necessarily converge. To see this notice that in Line 3.22 the second summation converges when $\alpha_k < 1$ (I.e., the protocol is non blocking), while the first summation converges if and only if the term $(1-\alpha)\delta$ is strictly less than 1. Hence, the success probability α_k must satisfy $\alpha_k < 1 - 1/\delta$, *i.e.*,

$$F_{k,t} = \sum_{\tau=1}^{\infty}(1-\alpha_k)^{\tau-1}\alpha_k\delta^t\sum_{j=0}^{\tau-1}\delta^j = \begin{cases} \delta^t\frac{1}{1+\delta(\alpha_k-1)} & \alpha_k > 1-1/\delta; \\ \infty & \text{otherwise.} \end{cases}$$

For the special case of 2 agents, the above analysis yielded the closed form $q_2 = \frac{-1+\sqrt{1+4\delta}}{2\delta}$, and therefore, $\alpha_2 = (1 - \frac{-1+\sqrt{1+4\delta}}{2\delta})$. Substituting $1-1/\delta$ from this expression we get:

$$\alpha_2 - 1/(1-\delta) = \frac{-3+\sqrt{1+4\delta}}{2\delta},$$

which is greater than 0 for every $1 < \delta < 2$. Thus, for 2 agents, we have extended the region of δ in which a unique symmetric equilibrium exists, to $0 < \delta < 2$.

Our analysis in this chapter is based on a preference order derived by the utility function. It does not tell us how what a user preference is in the case that two protocols have expected cost that diverges to infinity. Notice that the sum in equation (3.17) diverges for every $\delta > 2$; the transmission probability in equilibrium is always greater than $1/k$, and therefore the success probability α_k is at most $(1-1/k)^{k-1} < 1/2$, for $k \geq 2$.

In corollary 3.10 we show that the transmission probability in equilibrium q_k is at least $1/k$, and therefore α_k is at most $(1-1/k)^{k-1} \approx 1/e$ for large k. Therefore, in our analysis of protocols for k agents, we can rule out δ such that $\delta > e/(e-1)$.

In the next theorem we are able to derive some upper bound on the transmission probability, conditioned on having finite expected cost. However, the bound is not enough to show that the cost is indeed finite.

Theorem 3.27. *If there exists a symmetric time-independent, non-blocking equilibrium, in which every agent has a finite expected cost, then the transmission probability in equilibrium* *is such that*

$$q_k < \frac{1}{k-1}\frac{\delta}{(\delta-1)e}.$$

Proof. Recall the definitions of $\alpha_{k,t}, \beta_{k,t}$ (in equations (3.1), (3.2)). The time index t is redundant in the description of $\alpha_{k,t}, \beta_{k,t}$, since time independent strategies are considered. Hence we will write α_k, β_k instead. The expected cost

$$C_{k,t} = \alpha_k \cdot \Psi(t+1) + \beta_k C_{k-1,t+1} + (1-\alpha_k-\beta_k)C_{k,t+1}. \tag{3.28}$$

Using Lemma 3.26 we can remove the dependence on time in equation (3.28),

$$C_{k,t} = \alpha_k \cdot (1+\delta\Psi(t)) + \beta_k(1+\delta C_{k-1,t}) + (1-\alpha_k-\beta_k)(1+\delta C_{k,t}).$$

Rearranging we get

$$(C_{k,t}-\Psi(t))(1-\delta(1-(\alpha_k+\beta_k))) - \delta^{t+1} = \delta\beta_k(C_{k-1,t}-\Psi(t)) \ ,$$

and since $\delta^t > 0$

$$(C_{k,t}-\Psi(t))(1-\delta(1-(\alpha_k+\beta_k))) > \delta\beta_k(C_{k-1,t}-\Psi(t)) \ .$$

We replace expected cost with future expected cost and get that

$$\frac{F_{k-1,t}}{F_{k,t}} = \frac{C_{k-1,t}-\Psi(t)}{C_{k,t}-\Psi(t)} < \frac{(1-\delta(1-(\alpha_k+\beta_k)))}{\delta\beta_k} \ .$$

Replacing α_k, β_k as a function of q_k, we get an upper bound on $\frac{F_{k-1,t}}{F_{k,t}}$ as a function of q_k and k.

$$\frac{F_{k-1,t}}{F_{k,t}} < \frac{1-\delta(1-kq_k(1-q_k)^{k-1})}{\delta(k-1)q_k(1-q_k)^{k-1}} \ . \tag{3.29}$$

Let $g(q) = q(1-q)^{k-1}$, and let $f(x) = \frac{1-\delta(1-kx)}{\delta(k-1)x}$. The right hand side of inequality (3.29) equals $f(g(q_k))$. We show now an upper bound on $f(g(q))$ in the range $q \in [0,1]$. Note that $\text{argmax}_{q\in[0,1]}g(q) = 1/k$. Therefore, $g(q) \le \frac{1}{k-1}(1-\frac{1}{k})^k < \frac{1}{e(k-1)}$, for every $0 \le q \le 1$. For

$x < \frac{1}{e(k-1)}$,

$$\begin{aligned} f(x) &= \frac{1-\delta(1-kx)}{\delta(k-1)x} \\ &= \frac{k}{k-1} - \frac{\delta-1}{\delta(k-1)x} \\ &< \frac{k}{k-1} - \frac{\delta-1}{\delta(k-1)\frac{1}{e(k-1)}} \\ &< \frac{k}{k-1} - \frac{e(\delta-1)}{\delta} \end{aligned}$$

Using Lemma 3.8, we get that

$$q_k < \frac{1}{k-(k-1)(\frac{k}{k-1} - \frac{e(\delta-1)}{\delta})} = \frac{1}{k-1} \cdot \frac{\delta}{e(\delta-1)},$$

■

Unfortunately, although Theorem 3.27 provides an asymptotically optimal bound on the transmission probability, it does not tight enough to guarantee the convergence of the expected cost. Theorem 3.27 does not guarantee that the transmission probability q_k does not get as high as $\frac{1}{k-1} \cdot \frac{\delta}{e(\delta-1)}$. In this case, the success probability has an upper bound of $1 - 1/\delta$, since

$$\begin{aligned} \alpha_k &= \left(1 - \frac{1}{k-1} \cdot \frac{\delta}{e(\delta-1)}\right)^{k-1} && (3.30) \\ &< e^{-\frac{\delta}{(\delta-1)e}} && (3.31) \\ &\overset{(a)}{\leq} \frac{1}{1+\frac{\delta}{(\delta-1)e}} && (3.32) \\ &= \frac{(\delta-1)e}{(\delta-1)e+\delta} && (3.33) \\ &< \frac{(\delta-1)e}{(\delta-1)e+\delta e} && (3.34) \\ &= 1 - \frac{1}{\delta}, && (3.35) \end{aligned}$$

where inequality (a) follows from the known relation $e^{-x} < 1/(1+x)$, for every $x \neq -1$.

It remains open to show that either cost is finite, for some range of $\delta > 1$, or that no such equilibrium exists. If there is a range of $\delta > 1$ where the cost of the symmetric equilibrium is finite, then Theorem 3.27 suggests that when agents have exponentially growing costs we observe a phenomenon which resembles the altruism that characterized agents with deadline — its better to cooperate now, since the future is very costly.

3.6 Future Research Directions

Natural directions for future research include the following:

(i) Give a protocol for latency costs with expected linear cost (not only with high probability). Alternately, prove that no such protocol exists.

(ii) Strategic behavior when the number of agents pending is unknown.

(iii) Allow packets to be inserted over time.

(iv) Other congestion functions, not only all or nothing. E.g., the outcome of a collision is probabilistic and results in no packet being transmitted with probability p_0, and one random packet being successful with probability p_1. Another example would be where the probability of noise causing a packet to be dropped grows with the number of conflicting transmissions.

(v) Consider more general networks, not only a single link.

Perhaps the most interesting direction for future research is to consider generalizations of Wardrop equilibria, or other equilibrium notions from the field of congested networks. Previous work on strategic behavior for multi-commodity flow implicitly assumes a steady state where the (s_i, t_i) flow for agent i has an associated "flow rate", the number of gallons per minute or the bandwidth.

Our work above suggests another parameter for study, the flow duration, *i.e.*, agent i requires flow from s_i to t_i for a duration of d_i. E.g., the flow rate is the bandwidth required to transmit MPEG, the flow duration is the length of the movie. In this setting, it may be advisable to behave politely and allows others to transmit so as to get them out of the way. E.g., it may be advisable to leave home at 10:00 AM so as to avoid rush hour traffic.

Chapter 4

Socially Concave Games

We study a general sub-class of concave games, which we call socially concave games. We show that if each agent follows any no-external regret minimization procedure then the dynamics converges in the sense that both the average action vector converges to a Nash equilibrium and that the utility of each agent converges to her utility in that Nash equilibrium.

We show that many natural games are socially concave games. Specifically, we show that linear Cournot competition, linear resource allocation games, and atomic, splittable, congestion games, with affine latencies, are all socially-concave games, and therefore our convergence result applies to them. In addition, we show that a simple best response dynamic might diverge for linear resource allocation games, and is known to diverge for a linear Cournot competition, and for atomic, splittable congestion games. For the TCP congestion games we show that "near" the equilibrium these games are socially-concave, and using our general methodology we show convergence of specific regret minimization dynamics.

In this chapter we will be studying no-external regret dynamics in a general subclass of games. In a concave game the utility function of each agent is concave in her own action. Rosen [89] showed that a Nash equilibrium in pure strategies always exists in such games. We concentrate on a sub-class of concave games, which we call *socially concave games*. We show that many interesting games are socially-concave. including linear Cournot competition [77], linear resource allocation games [63], and atomic, splittable routing games with affine costs [23, 54, 92].[1] In Chapter 5 we show that a class of congestion avoidance

[1] In a linear Cournot competition multiple firms compete by setting their production levels, and the price

Expression	**Definition**
Γ	A general game.
$\Gamma^C, \Gamma^{\mathrm{R}}$	Linear Cournot game; Linear resource allocation.
$\Gamma^{\mathrm{TD}}, \Gamma^{\mathrm{RED}}$	Congestion avoidance games with Tail-Drop policy, and RED policy respectively.
N	The set of agents.
$\mathcal{R}_i$	An upper bound on the regret of agent i at time t.
S_i	The strategy space of agent i.
S	The product of all agents' strategy spaces.
$u_i(\cdot)$	The utility function of agent i.
$x^\tau = (x_1^\tau, \ldots, x_n^\tau)$	The action profile played at time τ of the repeated game.
$\hat{x}^t$	The average action profile at time t, *i.e.*, $\hat{x}^t = 1/t \sum_{\tau=1}^{t} x^\tau$.
$\mathrm{BR}_i(x)$	The best response of agent i to when the other agents play action profile x_{-i}.
$\lambda = (\lambda_1, \ldots, \lambda_n)$	Positive weights on the agents utilities, such that $\sum_{i \in N} \lambda_i u_i$ is a concave function. By definition 4.1, λ exists in every socially concave game.

Table 4.1: Notation in use.

games [65] are socially-concave "near" their equilibrium.

We derive a general convergence result, showing that if each agent follows a no-external regret procedure, then the dynamics, in any socially-concave game reaches an equilibrium. The convergence is both of the average action vector, and the average utility of the individual agents. On the other hand, we show that a best response dynamics might diverge for linear resource allocation games, and for atomic routing games, and it is known to diverge for linear Cournot competition [96].

For the TCP congestion avoidance setting we study two different games, depending on how the network handles overflow. The first is related to router's tail-drop policies, was proposed in [65], and studied in the context of competitive online algorithms [65, 10]. We also study a game motivated by router's policy of Random Early Discard (RED) [42]. In both cases, although the games are not socially-concave, we show that gradient based no-external regret procedures (such as [103]) guarantee the desired convergence.

4.1 Preliminaries

In this chapter we consider a one stage concave game $\Gamma = \{N, \{S_i\}_{i\in N}, \{u_i\}_{i\in N}\}$

We assume that the strategy space S_i is a closed, convex and bounded action set and that $S_i \subset \mathbb{R}^{m_i}$ for some $m_i \in \mathbb{N}$. Also, we assume that the utility functions u_i are twice differentiable and bounded from above by 1.

In his seminal paper, Rosen [89] considered the class of *concave games*, where every agent's utility function u_i is concave in her own action $s_i \in S_i$. In this chapter we consider a closely related class we denote *socially concave* games, which we now define.

Definition 4.1. *A game is* socially concave *if the following holds:*

- *A1 There exists a strict convex combination of the utility functions which is a concave function. Formally, there exists an n-tuple $(\lambda_i)_{i\in N}$, $\lambda_i > 0$, and $\sum_{i\in N}\lambda_i = 1$, such that $g(x) = \sum_{i\in N}\lambda_i u_i(x)$ is a concave function of x in the domain $S = \prod_{i\in N} S_i$.*
- *A2 The utility function of each agent i, is convex in the actions of the other agents, i.e., for every $s_i \in S_i$ the function $u_i(s_i, x_{-i})$ is convex in x_{-i} in the domain $S_{-i} = \prod_{j\in N, j\neq i} S_j$.*

For a socially concave game Γ, we denote by $\lambda(\Gamma)$, the set of all vectors λ for which property A1 holds. Next, we show that the class of socially concave games is a sub-class of concave games.

Lemma 4.2. *Let $\Gamma = \{N, \{S_i\}_{i\in N}, \{u_i\}_{i\in N}\}$ be a socially concave game. If the set S_i is compact for every $i \in N$, then Γ is a concave game.*

Proof. Fix an agent $i \in N$ and a vector $x_{-i} \in S_{-i}$. The function u_i can be written as

$$u_i(x_i, x_{-i}) = \frac{1}{\lambda_i}\left(\sum_{j\in N}\lambda_j u_j(x_i, x_{-i}) + \sum_{j\in N, j\neq i}\lambda_j(-1)u_j(x_i, x_{-i})\right) .$$

Following property (A1), the first term inside the parenthesis is a concave function of x and therefore also a concave function of x_i. Following property (A2), for every $j \in N, j \neq i$ the function $-u_j$ is concave in x_{-j}, and therefore also concave in x_i, as i belongs to the set of indices $-j$. Accordingly, the second term inside the parenthesis is also a concave function of x_i. The claim then follows. ∎

is a linear function of the overall production level. In a linear resource allocation game where the agent receive a share of the resource (e.g., bandwidth or a market share), as a function of their investment.

Rosen [89] showed that every concave game has a Nash equilibrium point in S. Therefore, a Nash equilibrium exists in every socially concave game with compact strategy sets.

4.2 Main Result

We show that for the class of socially concave games, no-external regret behavior leads to a Nash equilibrium. For every t, let $\hat{x}^t$ be the average of the n-tuples of strategies played up to time t, that is $\hat{x}^t = \frac{1}{t}\sum_{\tau=1}^{t} x^\tau$. Let $\hat{u}^t$ denote the average utility vector up to time t, that is, for every agent i, $\hat{u}_i^t = \frac{1}{t}\sum_{\tau=1}^{t} u_i(x^\tau)$.

Theorem 4.3 (Main Theorem). *Let $\Gamma = (N, (S_i)_{i\in N}, (u_i)_{i\in N})$ be an n-agents socially concave game, and let $\lambda \in \lambda(\Gamma)$, $\lambda = (\lambda_i)_{i\in N}$, be the constants of property A1 in the definition of socially concave games. If every agent i plays according to a procedure with external regret bound $\mathcal{R}_i(t)$, then at time t,*

(i) The average strategy vector $\hat{x}^t$ is an ϵ^t-Nash equilibrium, where $\epsilon^t = \frac{1}{\lambda_{\min}}\sum_{j\in N}\frac{\lambda_j \mathcal{R}_j(t)}{t}$ and $\lambda_{\min} = \min_{j\in N}\lambda_j$.

(ii) The average utility of agent i is close to her utility at $\hat{x}^t$, the average vector of strategies. Formally,

$$|\hat{u}_i^t - u_i(\hat{x}^t)| \le \frac{1}{\lambda_i}\sum_{j\in N}\frac{\lambda_j \mathcal{R}_j(t)}{t}.$$

Proof. The proof follows six steps:

Step I: By definition of regret minimization algorithms, for any period $\{1,\ldots,t\}$, each agent $i \in N$ has low regret to any action $s_i \in S_i$. Specifically, we can apply it to the best response action to $\hat{x}_{-i}^t = \frac{1}{t}\sum_{\tau=1}^{t} x_{-i}$, denoted $\mathrm{BR}_i(\hat{x}_{-i}^t) \in S_i$. Therefore,

$$\hat{u}_i^t = \frac{1}{t}\sum_{\tau=1}^{t} u_i(x^\tau) \ge \max_{s_i\in S_i}\frac{1}{t}\sum_{\tau=1}^{t} u_i\left(s_i, x_{-i}^\tau\right) - \mathcal{R}_i(t) \ge \frac{1}{t}\sum_{\tau=1}^{t} u_i\left(\mathrm{BR}_i(\hat{x}_{-i}^t), x_{-i}^\tau\right) - \frac{\mathcal{R}_i(t)}{t}. \quad (4.1)$$

Step II: Fix an action $y_i \in S_i$ of agent i. Property (A2) states that $u_i(y_i, x_{-i})$ is convex in its second argument x_{-i}, which implies,

$$\frac{1}{t}\sum_{\tau=1}^{t} u_i(y_i, x_{-i}^\tau) \ge u_i(y_i, \frac{1}{t}\sum_{\tau=1}^{t} x_{-i}^\tau) = u_i(y_i, \hat{x}_{-i}^t). \quad (4.2)$$

Step III: By definition of best response, for every n-tuple of strategies $y \in S$,

$$u_i(\mathrm{BR}_i(y_{-i}), y_{-i}) \geq u_i(y). \tag{4.3}$$

Step IV: Property (A1) states that for some $\lambda = (\lambda_i)_{i\in N}$, $g(x) = \sum_{i\in N} \lambda_i u_i(x)$ is concave. By the concavity of $g(x)$ we have that,

$$\sum_{i\in N} \lambda_i u_i(\hat{x}^t) = \sum_{i\in N} \lambda_i u_i \left(\frac{1}{t}\sum_{\tau=1}^{t} x^\tau\right) \geq \sum_{i\in N} \lambda_i \sum_{\tau=1}^{t} \frac{1}{t} u_i(x^\tau) = \sum_{i\in N} \lambda_i \hat{u}_i^t. \tag{4.4}$$

Step V: Combining inequalities (4.1)—(4.4) we get the following chain of inequalities:

$$\begin{aligned}
\sum_{i\in N} \lambda_i \hat{u}_i^t = \sum_{i\in N} \lambda_i \left(\frac{1}{t}\sum_{\tau=1}^{t} u_i(x^\tau)\right) &\overset{(a)}{\geq} \sum_{i\in N} \lambda_i \left(\frac{1}{t}\sum_{\tau=1}^{t} u_i\left(\mathrm{BR}_i(\hat{x}_{-i}^t), x_{-i}^\tau\right) - \frac{\mathcal{R}_i(t)}{t}\right) \\
&\overset{(b)}{\geq} \sum_{i\in N} \lambda_i \left(u_i(\mathrm{BR}_i(\hat{x}_{-i}^t), \hat{x}_{-i}^t) - \frac{\mathcal{R}_i(t)}{t}\right) \\
&\overset{(c)}{\geq} \sum_{i\in N} \lambda_i \left(u_i(\hat{x}^t) - \frac{\mathcal{R}_i(t)}{t}\right) \\
&\overset{(d)}{\geq} \sum_{i\in N} \lambda_i \left(\frac{1}{t}\sum_{\tau=1}^{t} u_i(x^\tau)\right) - \sum_{i\in N} \lambda_i \frac{\mathcal{R}_i(t)}{t},
\end{aligned}$$

where (a) follows from (4.1), (b) follows from (4.2) with $y_i = \mathrm{BR}_i(\hat{x}_{-i}^t)$, (c) follows from (4.3) with $y = \hat{x}^t$, and (d) follows from (4.4).

The above inequalities imply that

$$\left|\sum_{i\in N} \lambda_i u_i(\hat{x}^t) - \sum_{i\in N} \lambda_i u_i(\mathrm{BR}_i(\hat{x}_{-i}^t), \hat{x}_{-i}^t)\right| \leq \sum_{i\in N} \lambda_i \frac{\mathcal{R}_i(t)}{t},$$

since given any set of a_i's such that $a_1 \geq a_2 \geq a_3 \geq a_4 \geq a_5$ we have that $|a_3 - a_2| \leq |a_1 - a_5|$. Hence,

$$\sum_{i\in N} \lambda_i u_i(\hat{x}^t) \geq \sum_{i\in N} \lambda_i u_i(\mathrm{BR}_i(\hat{x}_{-i}^t), \hat{x}_{-i}^t) - \sum_{i\in N} \lambda_i \frac{\mathcal{R}_i(t)}{t}. \tag{4.5}$$

By definition, for every i, $u_i(\hat{x}^t) \leq u_i(\mathrm{BR}_i(\hat{x}^t_{-i}), \hat{x}^t_{-i})$. Therefore, For every $j \in N$,

$$\sum_{i \neq j} \lambda_i u_i(\hat{x}^t) \leq \sum_{i \neq j} \lambda_i u_i(\mathrm{BR}_i(\hat{x}^t_{-i}), \hat{x}^t_{-i}).$$

Combining this with (4.5) we get that for every $j \in N$,

$$\lambda_j u_j(\hat{x}^t) \geq \lambda_j u_j(\mathrm{BR}_j(\hat{x}^t_{-j}), \hat{x}^t_{-j}) - \sum_{i \in N} \frac{\lambda_i \mathcal{R}_i(t)}{t}.$$

Thus $\hat{x}^t$ is an ϵ^t-Nash equilibrium for $\epsilon^t = \frac{1}{\lambda_{\min}} \sum_{i \in N} \lambda_i \frac{\mathcal{R}_i(t)}{t}$.

Step VI: Similar to Step V, from inequalities (4.1), (4.2) and (4.3),

$$\begin{aligned}
\hat{u}_i^t = \left(\frac{1}{t} \sum_{\tau=1}^{t} u_i(x^\tau) \right) &\overset{(a)}{\geq} \frac{1}{t} \sum_{\tau=1}^{t} u_i \left(BR(\hat{x}^t_{-i}), x^\tau_{-i} \right) - \frac{\mathcal{R}_i(t)}{t} \\
&\overset{(b)}{\geq} u_i(\mathrm{BR}_i(\hat{x}^t_{-i}), \hat{x}^t_{-i}) - \frac{\mathcal{R}_i(t)}{t} \\
&\overset{(c)}{\geq} u_i(\hat{x}^t) - \frac{\mathcal{R}_i(t)}{t}
\end{aligned}$$

where (a) follows from (4.1), (b) follows from (4.2), and (c) follows from (4.3). Therefore, it follows that the average utility of agent i, $\hat{u}_i^t$, is at least her utility when the average strategy vector is played, $u_i(\hat{x}^t)$, minus her own average regret,

$$\hat{u}_i^t \geq u_i(\hat{x}^t) - \frac{\mathcal{R}_i(t)}{t}. \tag{4.6}$$

From inequality (4.4) we have that $\sum_{i \in N} \lambda_i u_i(\hat{x}^t) \geq \sum_{i \in N} \lambda_i \hat{u}_i^t$. Therefore,

$$\hat{u}_i^t - u_i(\hat{x}^t) \leq \frac{1}{\lambda_i} \sum_{j \neq i | j \in N} \lambda_j \left(u_j(\hat{x}^t) - \hat{u}_j^t \right) \leq \frac{1}{\lambda_i} \sum_{j \neq i | j \in N} \lambda_j \frac{\mathcal{R}_j(t)}{t} \leq \frac{1}{\lambda_i} \sum_{j \in N} \lambda_j \frac{\mathcal{R}_j(t)}{t}, \tag{4.7}$$

where the second inequality follows from (4.6).

Combining inequalities (4.6), and (4.7), we bound the difference between agent's i average utility and its utility at $\hat{x}^t$, *i.e.*,

$$|\hat{u}_i^t - u_i(\hat{x}^t)| \leq \frac{1}{\lambda_i} \sum_{j \in N} \lambda_j \frac{\mathcal{R}_j(t)}{t}.$$

■

The following are immediate consequences of Theorem 4.3. First, if every agent employs a no-regret algorithm, then the average strategy vector converges to a Nash equilibrium, and the average utility of each agent converges to her utility at that Nash equilibrium. Also, if every agent employs the *generalized infinitesimal gradient ascent* algorithm [103], which has regret $O(\sqrt{t})$, then after t steps the average strategy vector is an $O(n/\sqrt{t})$-Nash equilibrium and the average utility of each agent differs from her utility at that Nash equilibrium by at most $O(n/\sqrt{t})$, assuming that $\lambda_{\min}$ is bounded away from zero and that the utility's values are in the range $[0, 1]$.

A natural question to be asked is whether a stronger convergence result holds, where the daily action profile converges to Nash equilibrium, and not only the average action profile. Unfortunately the answer is negative. If the only requirement regarding the learning algorithm is that it guarantees no external-regret, then there exists pathological no-regret algorithms for which the daily actions profile does not converge, but rather oscillate between several work points.

Theorem 4.4. *When each agent employs some no external-regret algorithm, the daily action profile need not necessarily converge.*

Proof. We give a proof in Section 4.5.2, based on a routing game defined there. ■

4.3 Cournot Competition

Cournot competition [29] is a fundamental economic model used to describe competition between firms. The model considers multiple firms (oligopoly), which produce the same good. The main interaction between the firms is due to their influence on the good market price. Specifically, each firm decides on its production level (the quantity it produces from the good), and incurs an associated cost (which depends on the quantity, and may be different for different firms). The revenue of a firm is the product of its quantity and the market price, where market price depends on the aggregate quantity produced by all firms. Let us first define formally a Cournot competition.

Definition 4.5. *A* Cournot competition *is a game* $\Gamma^C = (N, (S_i)_{i\in N}, (c_i)_{i\in N}, p, (u_i)_{i\in N})$*, where* N *is the set of firms,* $S_i = \mathbb{R}_+$ *is the quantity firm* i *decides to produce,* $c_i : S_i \to \mathbb{R}_+$

is the cost of firm i to produce a quantity q_i, and $p : S \to \mathbb{R}_+$ is the price of the good (where $p(s)$ is a function of the aggregate quantity $s = \sum_{i \in N} s_i$), and $u_i : S \to \mathbb{R}$ is firm i utility such that $u_i(s) = s_i p(s) - c_i(s_i)$. A linear Cournot competition *is a Cournot competition where $p(s) = a - b(\sum_{i \in N} s_i)$, where a and b are some positive constants and the cost function c_i are convex.*

A well known result by Cournot (cf. [49, pp. 11]), is that if two firms participate in linear Cournot competition, and both play simultaneously the best response dynamics, then their joint play converges to equilibrium. Theocharis [96] showed that the best response dynamics fails to converge in linear Cournot competition, whenever there are four or more firms participating.

We show that in a linear Cournot competition, if each firm bases its production level on a no-external regret algorithm, then the firms' average utilities and quantities produced converges to the unique Nash Eequilibrium of the linear Cournot competition.

Lemma 4.6. *A linear Cournot competition is a socially concave game.*

Proof. To show that property A1 holds, consider the aggregate utility:

$$g(s) = \sum_{i \in N} u_i(s) = \sum_{i \in N} s_i(a - b(\sum_{j \in N} s_j)) - \sum_{i \in N} c_i(s_i),$$

which is concave in s, hence Assumption (A1) in Definition 4.1 holds. Since $u_i(s_i, s_{-i}) = a - b(\sum_{j \in N} s_j) - c_i(s_i)$ is a linear function in s_{-i} it is also convex in s_{-i}, hence, Assumption (A2) in Definition 4.1 holds as well. ∎

Since a linear Cournot competition is a socially concave game, Theorem 4.3 implies the convergence of the no-external regret dynamics.

Theorem 4.7. *In a linear Cournot competition, if each firm $i \in N$ employs a procedure with no external regret, then the average production level of every firm converges to its production level in a Nash equilibrium, and the average utility of each firm converges to its utility in that Nash equilibrium.*

Remark: We can show convergence for a larger class of Cournot competition. Namely, consider the case that the cost functions c_i are convex, $xp(x)$ is concave, and $p(x)$ is convex, where $x = \sum_{i \in N} s_i$. Since $p(x)$ is convex, it implies that $u_i(s)$ is convex in s_{-i}, and this satisfies Assumption (A2) in Definition 4.1. Since the function $g(s) = \sum_{i\in N} s_i p(x) - c_i(s_i) = xp(x) - \sum_{i\in N} c_i(s_i)$ is concave, property (A1) in Definition 4.1 holds. Therefore, this is a socially concave game, and Theorem 4.3 guarantee the convergence.

4.4 Linear Resource Allocation Games

In a resource allocation game [51, 63] n users share a communication link of capacity $C > 0$ (we assume without loss of generality $C = 1$). Let d_i denote the rate allocated to user i. We assume that user i receives a value $\varphi_i(d_i)$ if the allocated rate is d_i; we assume that this value is measured in monetary units.

Each user i submits a "bid" w_i to the network from a bid space $S_i = [0, 1]$. The network accepts these submitted bids and determines the share of capacity each user is allocated, according to an allocation function $M : S^n \to [0, C]^n$, mapping the bids to a feasible allocation (*i.e.*, for any $w \in S$ we have $\sum_{i\in N} M_i(s) \le 1$ and $M_i(w) \ge 0$ for every i). This makes the model a game Γ^{R} between the n users, $\Gamma^{\mathrm{R}} = (N, (S_i)_{i\in N}, (u_i)_{i\in N}, M, (\varphi_i)_{i\in N})$, where user i utility function is $u_i(w) = \varphi_i(M_i(w)) - w_i$ and φ_i is its value function.

Hajek and Gopalakrishnan [51] studied resource allocation games with the proportional allocation function

$$M_i(w) = \begin{cases} \frac{w_i}{\sum_{j\in N} w_j} & \sum_{j\in N} w_j > 0 \\ 0 & \text{otherwise,} \end{cases}$$

and showed that when the value function φ_i, of each user i is concave, a unique Nash equilibrium of the Γ^{R} exists.

We will concentrate on the following sub-class of resource allocation games.

Definition 4.8. *A* linear resource allocation game *is a resource allocation game* $\Gamma^{\mathrm{R}} = (N, (S_i)_{i\in N}, (u_i)_{i\in N}, M, (\varphi_i)_{i\in N})$ *such that* $\varphi_i(d_i) = \alpha_i d_i$ *and the allocation mechanism is proportional.*

Theorem 4.9. *In a linear resource allocation game, if every agent employs a procedure with no external regret, then the average action of the agents will converge to a Nash equilibrium, and the average utility of each agent will converge to her utility in that Nash equilibrium.*

Theorem 4.9 follows immediately from Theorem 4.3, once we establish in the following lemma that a linear resource allocation game is a socially concave game.

Lemma 4.10. *A linear resource allocation game is socially concave game.*

Proof. In a linear resource allocation game, the utility of agent i is

$$u_i(w) = \varphi_i(w) - w_i = \alpha_i M_i(w) - w_i = \begin{cases} \alpha_i \frac{w_i}{\sum_{j\in N} w_j} - w_i & \sum_{j\in N} w_j > 0 \\ 0 & \text{otherwise,} \end{cases}$$

To show property (A1) in Definition 4.1 holds, set $\lambda_i = \frac{\frac{1}{\alpha_i}}{\sum_{j\in N}\frac{1}{\alpha_j}}$, and consider the function

$$g(w) = \sum_{i\in N} \lambda_i u_i(w) = \begin{cases} \frac{1}{\sum_{j\in N}\frac{1}{\alpha_j}}\left(1 - \sum_{i\in N}\frac{w_i}{\alpha_i}\right) & \sum_{j\in N} w_j > 0 \\ 0 & \text{otherwise} \end{cases}$$

Notice that g is a linear function in the half open interval $(0,1]^n$, and therefore for every two points $w_1, w2$, and $\mu \in [0,1]$,

$$g(\mu w_1 + (1-\mu)w_2) = \mu g(w_1) + (1-\mu)w_2,$$

and for every $w \neq 0$,

$$g(\mu w + (1-\mu)0) = g(\mu w) = \mu(g(w) + (1-\mu)g(0),$$

and therefore, by definition, g is concave in the entire set S.

To show property (A2) in Definition 4.1 holds, we need to show that for every fixed w_i, $u_i(w_i, w_{-i})$, as a function of w_{-i} is a convex. When $w_i = 0$ we have that $u_i(w_i, w_{-i}) = 0$, for all $w_{-i} \in S_{-i}$, which a convex function.

For $w_i > 0$, note that $f(w_{-i}) = u(w_i, w_{-i}) + w_i = \frac{w_i}{w_i + \sum_{j\in N, j\neq i} w_j}$ is convex iff $u(w_i, w_{-i})$ is convex in w_{-i}. For a fixed $w_i > 0$, the function f can be written as a composition of two functions

$$f(x) = \ell(h(x)),$$

where

$$h(x) = \sum_{i=1}^{n-1} x_i,\ x \in \mathbb{R}_+^{n-1}\ ,\ \text{ and } \ell(y) = \frac{w_i}{w_i + y},\ y \in \mathbb{R}_+.$$

The function h is linear and, the function $\ell(y)$ is convex, as the inverse of a positive linear function. We therefore obtain that f is convex as a composition of a convex function h over a linear function ℓ (cf. [19, pp. 87]), and thus $u(w_i, w_{-i})$ is convex in x_{-i}. ∎

4.4.1 Divergence of the best response dynamics in resource allocation games

In the *best response dynamics*, every agent optimizes her decision for the next step assuming all other agents will play the same as they did in the previous step. Namely, at time t, agent i plays $x^t = \mathrm{BR}_i(x_{-i}^{t-1})$. Clearly, for this dynamics, if the joint vector of actions converges, then it must be that it converges to a Nash equilibrium. However, unlike the no regret dynamics, we show that the best response dynamics is not guaranteed to converge.

Consider an n-agent linear resource allocation game with $\gamma = 1$, where all n agents have identical utility for resource $\varphi_i(d_i) = d_i$. Namely, the utility of agent i, in terms of her bid and the bids of the other agents is,

$$u_i(x) = \frac{x_i}{\sum_{j\in N} x_j} - x_i.$$

The best response of agent $i \in N$ to $x_{-i} \in S_{-i}$ is $\max(0, \sqrt{\sum_{j\neq} x_j} - \sum_{j\neq} x_j) \in S_i$.

Theorem 4.11. *Consider a linear allocation game with identical utilities $\varphi_i(s_i) = s_i$. For $n \geq 4$, the best response dynamics does not necessarily converge.*

Proof. In equilibrium, each agent bids $x_i^{\mathrm{NE}} = \frac{n-1}{n^2}$. Consider the dynamics where initially all agents bid equally $x^0 \neq \frac{n-1}{n^2}$. Due to the symmetry, the best response dynamics would keep the bids of the agents equal. Let x^t be the bids of the agents time t (they are all identical), then

$$x^{t+1} = \sqrt{(n-1)x^t} - (n-1)x^t \tag{4.8}$$

Let ϵ^t be the difference between the agents bids at time t and the equilibrium, *i.e.*, $\epsilon^t = x^t - \frac{n-1}{n^2}$. By substituting x^t by $\frac{n-1}{n^2} + \epsilon^t$ in equation 4.8 we get

$$\frac{n-1}{n^2} + \epsilon^{t+1} = \sqrt{(n-1)(\frac{n-1}{n^2} + \epsilon^t)} - (n-1)(\frac{n-1}{n^2} + \epsilon^t) \tag{4.9}$$

We will show that the sequence $|\epsilon^t|$ does not converge to zero. We can describe ϵ^{t+1} as a function of ϵ^t,

$$\begin{aligned}
\epsilon^{t+1} &= \sqrt{\left(\frac{n-1}{n}\right)^2 + \epsilon^t(n-1)} - \left(\frac{n-1}{n}\right)^2 - \frac{n-1}{n^2} - \epsilon^t(n-1) \\
&= \sqrt{\left(\frac{n-1}{n}\right)^2 + \epsilon^t(n-1)} - \sqrt{\left(\frac{n-1}{n}\right)^2} - \epsilon^t(n-1) \\
&= \frac{\epsilon^t(n-1)}{\sqrt{\left(\frac{n-1}{n}\right)^2 + \epsilon^t(n-1)} + \sqrt{\left(\frac{n-1}{n}\right)^2}} - \epsilon^t(n-1) \\
&= \epsilon^t(n-1)\left(\frac{1}{\sqrt{\left(\frac{n-1}{n}\right)^2 + \epsilon^t(n-1)} + \sqrt{\left(\frac{n-1}{n}\right)^2}} - 1\right).
\end{aligned}$$

For $\epsilon^t \in [0, 3\frac{n-1}{n^2}]$ we have that

$$\epsilon^{t+1} \leq \epsilon^t(n-1)\left(\frac{1}{3\frac{n-1}{n}} - 1\right) = \epsilon^t\frac{3-2n}{3},$$

and for $n \geq 4$ we have $\epsilon^{t+1} \leq -(5/3)\epsilon^t < 0$.

For $\epsilon^t \in [-(7/16)\frac{n-1}{n^2}, 0]$ we have that

$$\epsilon^{t+1} \geq \epsilon^t(n-1)\left(\frac{1}{(7/4)\frac{n-1}{n}} - 1\right) = \epsilon^t\frac{7-4n}{7},$$

and for $n \geq 4$ we have $\epsilon^{t+1} \geq -(9/7)\epsilon^t > 0$. This implies that $|\epsilon^t|$ does not converge to zero. ∎

4.5 Atomic, Splittable Routing Games

Atomic routing games model agents behavior in a congested network, each of which wishes to route her traffic from a source to a destination, while minimizing her own latency.

An instance of an atomic, splittable routing game, is a triple (G, r, c), where G is a directed graph with (not necessarily distinct) source vertices $\{s_1, \ldots, s_n\}$, and sink vertices $\{t_1, \ldots, t_n\}$; r is a vector indexed by source-sink pairs, where agent i must route r_i units of traffic from s_i to t_i; and c is a vector of cost functions, one for each edge of G. It is standard to assume that each cost function c_e is non-negative, non-decreasing and semi-convex, *i.e.*, the weighted cost of a flow f, through an edge e, $fc_e(f)$ is convex (see [91]).

In order for such a game to become socially concave, we will need to further assume that c_e is concave, for every $e \in E$. The class of cost functions for which the above requirements hold includes among others, the class of affine cost functions, and more generally, every cost function of the form $c_e(f_e) = f_e^\gamma$ for $0 \leq \gamma \leq 1$.

For an instance (G, r, c), a feasible flow comprises n non-negative vectors $f^1, \ldots, f^n$, where f^i is defined on the $s_i - t_i$ paths $\mathcal{P}_i$ of G and satisfies $\sum_{P \in \mathcal{P}_i} f_P = r_i$. For a flow f, $f_e = \sum_{i=1}^n \sum_{P \in \mathcal{P}_i | e \in P} f_P$ denotes the total flow on edge e. The cost $c_P(f)$ of a path P with respect to a flow f is the sum $\sum_{e \in P} c_e(f_e)$ of the costs of its edges. The cost $C_i(f)$ to agent i is defined by $\sum_{P \in \mathcal{P}_i} c_P(f) f_P^i$, or equivalently, $\sum_{e \in E} c_e(f_e) f_e^i$, where f_e^i is defined by $\sum_{P \in \mathcal{P}_i | e \in P} f_P$.

A flow f is at Nash equilibrium if for each i, f^i minimizes $C_i(f)$ when the other flows $\{f^j\}_j \neq i$ are held fixed.

Lemma 4.12. *An atomic, splittable routing game (G, r, c), with cost functions that are both concave, and semi-convex, is a socially concave game.*

Proof. To show property (A1) holds, we need to show that $\sum_{i \in N} C_i(f) = \sum_{e \in E} c_e(f_e) f_e$ is a convex function[2]. The function $c_e(f_e) f_e$ is convex by assumption, thus, $\sum_{i \in N} C_i(f)$ is a convex function of f.

To show property (A2) holds we need to show that $C_i(f^i, f^{-i})$ is a concave function in its second argument. (*I.e.*, the utility $-C_i(f)$ is convex in f^{-i}). Let f_i be a fixed feasible $s_i - t_i$ flow of rate r_i. The cost for agent i as a function of the other flows is

$$C_i(f^i, f^{-i}) = \sum_{P \in \mathcal{P}_i} c_P(f) f_P = \sum_{e \in E} c_e(f_e^i + \sum_{j \neq i} f_e^j) f_e^i.$$

Where, for every edge $e \in E$, the function $g : f^{-i} \to \mathbb{R}$, defined by $g(f^{-i}) = c_e(f_e^i + \sum_{j \neq i} f_e^j) f_e^i$ is a concave function in f_{-i}, as a composition of a concave function c_e, with an affine transformation. Thus, $C_i(f_i, f_{-i})$ is a concave function in its second argument as the sum of concave functions. ∎

Theorem 4.13. *In an atomic, splittable, routing game, with cost functions that are both concave, and semi-convex, if every agent employs a procedure with no external regret, then*

[2] Notice that when cost(disutility) is considered instead of utility, we need to replace *convex* with *concave* in property (A1), and *concave* with *convex* in property (A2).

the daily average flow vector of the agents will converge to a Nash equilibrium, and the average cost of each agent will converge to its cost in that Nash equilibrium.

Remark 4.14. *In the previous examples for socially concave games (Cournot oligopoly, and resource allocation games), attaining no-regret could be done relatively straight forward, e.g., by using simple gradient based methods[103]. While it is possible to employ such methods in a direct way in the context of routing problems, the complexity of such an algorithm would be $O(\mathcal{P}_i)$, which could grow exponentially in the size of the graph G. Fortunately, efficient no-regret algorithms for online routing exist [64, 61]. Moreover, some of them guarantee no-regret even in the multi-armed bandit model [13, 1], where the feedback an agent receives after every time step consists only of her own cost at that step, and not of the entire loads in the network (see Section 2.2 in Chapter 2).*

4.5.1 The best response dynamics in atomic, splittable, routing games

In this section we show that generally, the simultaneous best-responds dynamics need not converge in a splittable routing game. For this matter we exhibit a simple example of a splittable routing game with $n \geq 3$ agents.

Consider a routing game played over a graph G, with 2 parallel edges. That is, there are two vertices in the graph, s, and t, which serve as a source and destination for $n \geq 3$ agents; there are two directed edges e_1, e_2, with end points at s, and t, and in the direction of t. The latency function of both edges is linear, that is, $c(x) = x$; each agent i has a rate $r_i = 1$.

This game has a unique Nash equilibrium in which every agent splits his load equally between the two edges. To show this assume that every agent but some fixed agent $i \in N$, indeed splits her flow equally between the two edges. Let $f_{i,1}$ denote the flow of agent i through one edge (the strategy space of an agent is the one dimension simplex since $f_{i,2} = 1 - f_{i,1}$). The cost for i as a function of f_i is

$$C_i(f_i, f_{-i}) = f_{i,1}\left(\frac{n-1}{2} + f_{i,1}\right) + (1 - f_{i,1})\left(\frac{n-1}{2} + 1 - f_{i,1}\right),$$

which attains its minimum at $f_{i,1} = 1/2$.

On the other hand, consider a situation where at some stage, every agent routes her total flow through the same edge e_1. In this case, the cost of an agent i as a function of his

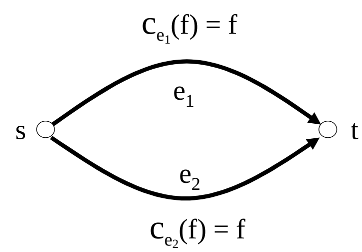


Figure 4.1: When two players with one unit of flow each play an atomic splittable routing game on this graph instance, alternating between e_1, and e_2 results in a sequence with no-external regret.

flow through e_1 would be

$$C_i(f_i, f_{-i}) = f_{i,1}(n + f_{i,1}) + (1 - f_{i,1})(1 - f_{i,1}),$$

which attains its minimum at $f_{i,1} = 0$, when $n \geq 3$. Namely, her best response is to route her complete flow on the unoccupied edge e_2. Of course, this means that the actions of the agents would oscillate between sending their complete flow through e_1, and sending the complete flow through e_2 indefinitely.

Notice that the average cost of an agent converges to n, and not to her Nash equilibrium cost which is $n/2$.

4.5.2 Non-Convergence of the daily action profile

We now provide an example of a 2 agents socially concave game, in which both agents act in accordance with a no external regret algorithm, and yet their daily action profile does not converge to a point.

Consider an atomic splittable routing game, played on G defined above. Suppose there are 2 agents, Alice and Bob, each with one unit of flow to transfer from s to t. Alice employs the following algorithm: on the first day, Alice sends her flow through the edge e_1. Then, while Bob is never sending any flow on the same edges as she does, she is sending her flow on e_2 on even days and e_1 on odd days. Once Bob is sending even a tiny fraction of flow on the edge as she is, she starts running the no regret algorithm of Kalai and Vempala from [64]. Suppose further that Bob runs a similar algorithm, only that Bob initially sends his flow on e_2.

In the resulting sequence, Alice always routes through e_1 on odd days, and through e_2

on even days, and Bob does the opposite. It is easy to see that Both Alice, and Bob have no-external regret. Indeed, Alice's average cost in the resulting sequence is 1. In contrast, if Alice where to play any fixed daily action, her average cost would increase.

To see this let $f^* = (f_1^*, f_2^*)$ be a fixed flow, where f_1^* is the fraction of flow Alice routes through e_1, and $f_2^* = 1 - f_1^*$ is the fraction of flow she routes through e_2. Her average cost for playing f^* on every day would be

$$\begin{aligned}\frac{1}{2}\left(f_1^* f_1^* + (1-f_1^*)(1+1-f_1^*)\right) + \frac{1}{2}\left(f_1^*(1+f_1^*) + (1-f_1^*)(1-f_1^*)\right) &= 2({f_1^*}^2 - f_1^*) + \frac{3}{2} \\ &\geq \frac{5}{4},\end{aligned}$$

where the inequality follows from the fact that ${f_1^*}^2 - f_1^* \geq -1/4$, for every $f_1^* \in [0,1]$.

The fact that Bob has no-external regret follows from a similar argument. We note that Alice, and Bob average action profile converges to the Nash equilibrium of the game, and that their average cost converges to the costs at this Nash Equilibrium. Nevertheless, the daily action profile does not converge.

4.6 A Sufficient Condition for the Uniqueness of Equilibrium in Socially Concave Games

In this section we show a sufficient condition for the uniqueness of equilibrium in socially concave games. It was already mentioned that the existence of an equilibrium in socially concave games is guaranteed due to Rosen [89], but generally, a socially concave game may admit multiple equilibria points. For example, the set of zero-sum games is trivially contained in the set of socially concave games, and zero-sum games usually consist of multiple equilibria.

In contrast, we now show, that if either property (A1) holds with strict concavity, or property (A2) holds with strict convexity, then the equilibrium is unique.

Theorem 4.15. *Let $\Gamma = \{N, \{S_i\}_{i\in N}, \{u_i\}_{i\in N}\}$ be a socially concave game. If each utility function u_i is differentiable, the sets S_i are compact for every $i \in N$, and either property (A1) holds with strict concavity, or property (A2) holds with strict convexity, then a Nash equilibrium is unique.*

Rosen[89] shows that for a concave game in which the utility functions satisfy an additional concavity requirement, he calls *diagonal strict concavity*, the Nash equilibrium point is unique. To explain what diagonal strict concavity means, we define a function $\sigma : \prod_{i \in N} S_i \times \mathbb{R}^n_+ \to \mathbb{R}$ to be a non-negative sum of the agents' utility functions, weighted by $\lambda \in \mathbb{R}^n_+$: $\sigma(s, \lambda) = \sum_{i \in N} \lambda_i u_i(s)$. For each fixed $\lambda \in \mathbb{R}^n_+$ a related mapping $g(s, \lambda)$ in terms of the gradients $\nabla_i u_i(s)$ is defined by

$$g(s, \lambda) = \begin{bmatrix} \lambda_1 \nabla_1 u_1(s) \\ \lambda_2 \nabla_2 u_2(s) \\ \vdots \\ \lambda_n \nabla_n u_n(s) \end{bmatrix}. \tag{4.10}$$

Definition 4.16 (Diagonal strict concavity from [89])**.** *The function σ will be called diagonally strictly concave for a joint strategy space S, and a fixed $\lambda \in \mathbb{R}^n_+$ such that $\lambda_i \geq 0$, for every $1 \leq i \leq n$, if for every two distinct strategy profiles, $s^0, s^1 \in S$, we have*

$$(s^1 - s^0)^T g(s^0, \lambda) + (s^0 - s^1)^T g(s^1, \lambda) > 0$$

We next show a sufficient condition for which σ, defined over the utility functions of a socially concave game is diagonally strictly concave.

Lemma 4.17. *Consider a socially concave game $\Gamma = \{N, \{S_i\}_{i \in N}, \{u_i\}_{i \in N}\}$. If for every $i \in N$, the utility function u_i is differentiable, and the set S_i is compact, and in addition, either property (A1) holds with strict concavity, or property (A2) holds with strict convexity, then the function σ is diagonally strictly concave.*

Proof. Let us denote by $\nabla f(s)$ the gradient with respect to s of a real function f, and let us denote by $\nabla_i f(s)$ the gradient with respect to s_i of $f(s)$. Let $s^T \in \mathbb{R}^{1 \times n}$ denote the transpose of a vector $s \in \mathbb{R}^{n \times 1}$.

For every differentiable concave function f, and every s^1, s^0 in the domain S of f, the gradient inequality holds (see [19]):

$$f(s^1) \leq f(s^0) + (s^1 - s^0)^T \nabla f(s^0), \tag{4.11}$$

with strict inequality in case f is strictly concave. Equivalently

$$f(s^0) \leq f(s^1) + (s^0 - s^1)^T \nabla f(s^1). \tag{4.12}$$

Summing inequalities (4.11) and (4.12) we obtain

$$0 \leq (s^1 - s^0)^T \nabla f(s^0) + (s^0 - s^1)^T \nabla f(s^1), \tag{4.13}$$

with strict inequality in case that f is strictly concave. Similarly, for a convex function f

$$0 \geq (s^1 - s^0)^T \nabla f(s^0) + (s^0 - s^1)^T \nabla f(s^1). \tag{4.14}$$

Assume first that property (A1) holds with strict concavity, *i.e.*, there exists $\lambda > 0$ such that the function $\sum_{i \in N} \lambda_i u_i(s)$ is strictly concave in the domain $s \in S$. By inequality (4.13) we have that

$$\begin{aligned} 0 \quad < \quad & (s^1 - s^0)^T \left(\nabla_1 \sigma(s^0, \lambda), \nabla_2 \sigma(s^0, \lambda), \cdots, \nabla_n \sigma(s^0, \lambda)\right) + & (4.15) \\ & (s^0 - s^1)^T \left(\nabla_1 \sigma(s^1, \lambda), \nabla_2 \sigma(s^1, \lambda), \cdots, \nabla_n \sigma(s^1, \lambda)\right), & (4.16) \end{aligned}$$

where $\nabla_i \sigma(s^0, \lambda)$ is the gradient of σ with respect to s_i, at the point (s^0, λ).

By replacing $\nabla_j \sigma(s, \lambda) = \sum_{i \in N} \nabla_j \lambda_i u_i(s)$ we obtain

$$0 < \sum_{i \in N} \left((s^1 - s^0)^T \begin{bmatrix} \lambda_1 \nabla_1 u_i(s^0) \\ \lambda_2 \nabla_2 u_i(s^0) \\ \vdots \\ \lambda_n \nabla_n u_i(s^0) \end{bmatrix} + (s^0 - s^1)^T \begin{bmatrix} \lambda_1 \nabla_1 u_i(s^1) \\ \lambda_2 \nabla_2 u_i(s^1) \\ \vdots \\ \lambda_n \nabla_n u_i(s^1) \end{bmatrix} \right) \tag{4.17}$$

$$= \sum_{i \in N} \left((s^1 - s^0)^T \begin{bmatrix} \lambda_1 \nabla_1 u_i(s^0) \\ \lambda_2 \nabla_2 u_i(s^0) \\ \vdots \\ \lambda_{i-1} \nabla_{i-1} u_i(s^0) \\ 0 \\ \lambda_{i+1} \nabla_{i+1} u_i(s^0) \\ \vdots \\ \lambda_n \nabla_n u_i(s^0) \end{bmatrix} + (s^0 - s^1)^T \begin{bmatrix} \lambda_1 \nabla_1 u_i(s^1) \\ \lambda_2 \nabla_2 u_i(s^1) \\ \vdots \\ \lambda_{i-1} \nabla_{i-1} u_i(s^1) \\ 0 \\ \lambda_{i+1} \nabla_{i+1} u_i(s^1) \\ \vdots \\ \lambda_n \nabla_n u_i(s^1) \end{bmatrix} \right) \tag{4.18}$$

$$+ \left((s^1 - s^0)^T \begin{bmatrix} \lambda_1 \nabla_1 u_1(s^0) \\ \lambda_2 \nabla_2 u_2(s^0) \\ \vdots \\ \lambda_n \nabla_n u_n(s^0) \end{bmatrix} + (s^0 - s^1)^T \begin{bmatrix} \lambda_1 \nabla_1 u_1(s^1) \\ \lambda_2 \nabla_2 u_2(s^1) \\ \vdots \\ \lambda_n \nabla_n u_n(s^1) \end{bmatrix} \right) \tag{4.19}$$

For every $i \in N$, the term inside the parenthesis in line (4.18) is non-positive, as follows from the fact u_i is convex in its parameter s_{-i} (property (A2)), and inequality (4.14). Consequently, the term in line (4.19) must be positive, and therefore diagonal strict concavity holds.

For the second case, we assume that property (A2) holds with strict convexity. In this case, the strict inequality in line 4.15 is replaced with a non-strict inequality sign, however, the term inside the parenthesis in line (4.18) becomes strictly negative. Again, it follows that the term in line (4.19) must be positive, and diagonal strict concavity follows. ∎

In Sections 4.3, to 4.5 several examples of socially concave games are given. In every one of these classes, either property A1 holds with strict concavity (for atomic splittable routing games, and Cournot oligopoly), or property A2 holds with strict convexity (resource allocation). Therefore, we can assert that equilibrium in the atomic splittable games, and the Cournot oligopoly game is unique[3]. We note however, that uniqueness of equilibrium in these games is already known (see [51] for the uniqueness of Nash equilibrium in resource allocation games, [77] for Cournot oligopoly, and [6] for atomic splittable routing).

Corollary 4.18. *A unique Nash equilibrium exists in any of the following games*

(i) Linear Cournot oligopoly.

(ii) Atomic splittable routing, with costs that are simultaneously semi-convex, and concave.

[3]For the resource allocation game in Section 4.4 the function we gave in order to show property A1 holds was not twice differentiable at the origin, and therefore Rosen's uniqueness result does not follow.

Chapter 5

Strategic Protocols for Congestion Avoidance

In the previous chapter we have introduced the class of socially concave games, and showed a general convergence property of no-regret dynamics when played over games in this class. In this chapter we continue to investigate no-regret dynamics in the context of analyzing protocols for congestion avoidance.

5.1 Congestion Avoidance Games

In this section we present an application of Theorem 4.3 in the design of protocols for congestion control. We consider multiple connections sharing a network path, with a common bottleneck. Time is divided into successive rounds, *i.e.*, time is discrete and events happen only at time $t = 1, 2, 3, \ldots$. It is assumed that all connections have the same round trip time. At the start of a round each connection transmits a window of packets and at the end of a round each connection receives a feedback with the number of packets that were actually delivered. Pending data is always available for sending at every source.

Each connection is associated with a single selfish agent. An agent benefits from delivering packets, and suffers a penalty for dropped packets, due to retransmission delays and overhead. We view the actions of the agents as the 'load' they introduce (or alternatively, the bandwidth they consume). Let x_i^t denote the load imposed by the i'th agent at time t, and denote by b_i^t her actual load *i.e.*, the fraction of packets that were not dropped. Agent i utility is $b_i^t - \alpha_i(x_i^t - b_i^t)$, namely, α_i is a parameter that reflects i's cost for losing a packet (or more precisely, one 'unit' of bandwidth).

The shared bottleneck is controlled by a router with finite capacity C assumed to be 1 unit of bandwidth. The router receives packets from each connection and decides to forward or discard each packet [1]. Different scheduling policies would share the capacity in different ways. And generally, a scheduling policy maps transmission rate vectors $x = (x_i)_{i \in N} \in \mathbb{R}^n_+$ to feasible bandwidth allocations $\{b \mid \sum_i b_i \leq C = 1\}$. Clearly, no mechanism can assign more than the link capacity, but some mechanisms might be more restrictive. We consider several scheduling policies:

1. *Tail drop* (TD). Load is accepted while the channel is not full; packets are dropped when the total transmission exceeds the capacity.

2. *Random early discard (RED).* Packets are randomly dropped with a dropping probability that increases with the offered load.

The policies, TD and RED, to be formally defined in later sections, when combined with a vector $\alpha = (\alpha_i)_{i \in N}$, admits a game $\Gamma^{\mathrm{TD}}, \Gamma^{\mathrm{RED}}$, respectively. Unfortunately, none of these games is socially concave, and therefore Theorem 4.3 cannot be applied directly. Luckily, in a region near the Nash equilibrium of every such game, these games becomes a socially concave game. We use this fact, to show that the *generalized infinitesimal gradient ascent* (GIGA) procedure attains the convergence properties guaranteed for socially concave games in Theorem 4.3.

Definition 5.1 (GIGA). *For agent i with a utility function u_i, and strategy space S_i, GIGA sets i's action at time t, x_i^t in the following method:*

$$y_i^t = x_i^{t-1} + \eta_t \frac{\partial}{\partial x_i} u_i(x^{t-1}) \ ; \ x_i^t = \pi(S_i, y_i^t)$$

where $\pi(S_i, y_i^t)$ is the projection of y_i^t into the set S_i,[2] *and η_t is a learning rate where we assume that: (i) η_t is non-increasing in t, i.e., $\eta_t \geq \eta_{t+1}$, (ii) η_t vanishes, i.e., for every $\epsilon > 0$ there is a time t_ϵ such that $\eta_{t_\epsilon} < \epsilon$, and (iii) that the sum of η_t diverges, i.e., for any*

[1] No queueing is assumed in our model. The technical difficulty with introducing a queue in our model is that a queue is essentially a state, and this will be a miss-match to both the repeated nature of the game and the regret minimization.

[2] *In our setting the projection only means that if $y_i^t < 0$ then we set $x_i^t = 0$.*

ρ there is at time t_ρ, such that $\sum_{\tau=1}^{t_\rho} \eta_\tau > \rho$.[3]

Theorem 5.2 (From [103]). *If $\eta_t = t^{-1/2}$, the regret of the Greedy Projection algorithm is*

$$\mathcal{R}(T) \leq \frac{\|S_i\|\sqrt{T}}{2} + \left(\sqrt{T} - \frac{1}{2}\|\nabla u_i\|^2\right),$$

where $\|S_i\| = \max_{x_1,x_2 \in S_i} \|x_1 - x_2\|$, and $\|\nabla u_i\| = \max_{x_i \in S_i, t \in \{1,2,\dots\}} \|\nabla u_i(x, x_{-i}^t)\|$.

Since $\|S_i\|$ for every i is a constant, and $\|\nabla u(\cdot, x_{-i})\|$ is a constant for $x_{-i} \in S_{-i}$, we have that the regret bounds of GIGA are $O(\sqrt{T})$.

Assuming that an agent learns her own utility as a feedback in each of the games Γ^{TD} and Γ^{RED}, it is possible for her to compute her own gradient $\frac{\partial}{\partial x_i} u_i(x^t)$, at time t, and calculate her next action in the GIGA procedure accordingly[4].

To show convergence of the GIGA algorithm to Nash equilibrium, we observe that for both games $\Gamma^{\text{TD}}, \Gamma^{\text{RED}}$, a subset of the agents joint strategy space $S' \subset S$, exists, such that a game restrained to S', is a socially concave game. We then show that if the agents are playing GIGA for a sufficiently long period, then their joint action profile is in the set S', for all subsequent steps. Since GIGA guarantees no-regret, we are then able to use Theorem 4.3 to show the convergence result.

5.2 Tail Drop

When a router employs a *tail drop* policy, packets are accepted as long as the overall load does not exceed the link capacity. This is modeled as follows. Define $\mathcal{S}(x) = \sum_{i \in N} x_i$ to be the sum of all transmissions levels. While $\mathcal{S}(x) \leq 1$, every agent gets her transmission rate, x_i. But, when $\mathcal{S}(x) > 1$ agent i gets only a share of the capacity, proportional to her transmission rate. This implies the following utility function:

$$u_i^{\text{TD}}(x) = \begin{cases} x_i & \mathcal{S}(x) \leq 1 \\ \frac{x_i}{\mathcal{S}(x)} - \alpha_i \left(x_i - \frac{x_i}{\mathcal{S}(x)}\right) & \mathcal{S}(x) > 1 \end{cases}$$

Let $\alpha_{\min} = \min_{j \in N} \alpha_j$; $\alpha_{\max} = \max_{j \in N} \alpha_j$. We assume that the penalty per packet loss

[3] *In case that the derivative $\frac{\partial}{\partial x_i} u_i(x^{t-1})$ is not continuous we define it to be the limit of the derivatives $x' < x$, which are always well defined in our setting.*

[4] In [65], a weaker feedback model is assumed, where after each step an agent receives a *binary feedback*, that tells him whether a congestion occurred or not.

parameters of the agents are bounded as follows, $\frac{3}{n-1} \leq \alpha_{\min} \leq \alpha_{\max} \leq 1$. We also assume that a single agent load never exceeds the channel capacity, *i.e.*, $x_i \in [0,1]$.

One can verify that the game $\Gamma^{\mathrm{TD}} = (N, ([0,1])_{i\in N}, (u_i^{\mathrm{TD}})_{i\in N})$ is not a socially concave game. However, a slight modification of Γ^{TD} results in a socially concave game, $\Gamma^{\mathrm{TD}'}$. The game $\Gamma^{\mathrm{TD}'}$ consists of the same set of agents and the same strategy space as the game Γ^{TD}. However, the utility function of an agent i is modified to

$$q_i(x) = \frac{x_i}{\mathcal{S}(x)} - \alpha_i \left(x_i - \frac{x_i}{\mathcal{S}(x)} \right).$$

Lemma 5.3. *The game* $\Gamma^{\mathrm{TD}'} = (N, ([0,1])_{i\in N}, (q_i)_{i\in N})$*, is a socially concave game.*

Proof. The utility function q_i is equivalent to a utility function of a resource allocation game, with proportional allocation mechanism, *i.e.*, $c = 1$. Hence, the same arguments made in the proof of Lemma 4.10 show that $\Gamma^{\mathrm{TD}'}$ is a socially concave game. ∎

Theorem 4.3 does not apply to Γ^{TD}, as it is not a socially concave game. Even so, using the fact that the game $\Gamma^{\mathrm{TD}'}$ is a socially concave game, we can show that the GIGA dynamics (*i.e.*, when all agents act according to the GIGA procedure), attains similar convergence properties as general no-regret dynamics in socially concave games.

Theorem 5.4. *Assuming there are at least $n \geq 4$ agents in the game, if every agent in a tail-drop game Γ^{TD} plays according to the GIGA procedure, then the average strategy vector will converge to a Nash equilibrium and the average utility of each agent will converge to her utility at that Nash equilibrium.*
Furthermore, if every agent runs GIGA with $\eta_t = 1/\sqrt{t}$, then at every time $t > t^*$, where $t^* = O(n^4)$,

(i) The average profile of actions will be an ϵ-Nash equilibrium, where $\epsilon = O(\frac{1}{\sqrt{t}})$

(ii) The average utility of each agent will differ from her utility at that Nash equilibrium by at most $O(\frac{1}{\sqrt{t}})$.

The main step in the proof of Theorem 5.4 is to show that after sufficient number of time steps the agents total offered load is always greater than the channel capacity.

Lemma 5.5. *Assuming there are at least $n \geq 4$ agents in the game, if every agent in a tail-drop game Γ^{TD} plays according to the GIGA procedure, then there exists a time t^* such that $\mathcal{S}(x^t) > 1$ for every $t > t^*$. Furthermore, if the learning rate $\eta_t = \frac{1}{\sqrt{t}}$, then $t^* = O(n^4)$.*

Proof. Let $\Delta_i^t(x) = \frac{\partial}{\partial x_i} u_i^{\mathrm{TD}}(x^{t-1})$ and let $\Delta(x) = \sum_{i \in N} \Delta_i(x)$. The proof proceeds in a number of steps.

Step I: *If x is such that $0 \le \mathcal{S}(x) < 1 + \epsilon$, where $\epsilon = (1 - \frac{2}{n+1})\alpha_{\min} - \frac{2}{n+1}$, then $\Delta(x) > 1$.*

First notice that following our assumption that $\alpha_{\min} \ge 3/(n-1)$,

$$\epsilon = (1 - \frac{2}{n+1})\alpha_{\min} - \frac{2}{n+1} > (1 - \frac{2}{n+1})\frac{3}{n-1} - \frac{2}{n+1} = \frac{1}{n+1} > 0.$$

Now, if $\mathcal{S}(x) \le 1$ then $\Delta(x) = n$ and the claim follows. Otherwise, $1 < \mathcal{S}(x) < 1 + \epsilon$, and in this case,

$$\begin{aligned}
\Delta(x) &= \sum_{i \in N} \left((1 + \alpha_i) \frac{\mathcal{S}(x) - x_i}{(\mathcal{S}(x))^2} - \alpha_i \right) && (5.1)\\
&\ge \sum_{i \in N} \left((1 + \alpha_{\min}) \frac{\mathcal{S}(x) - x_i}{(\mathcal{S}(x))^2} - 1 \right) && (5.2)\\
&= \frac{(1 + \alpha_{\min})(n-1)}{\mathcal{S}(x)} - n && (5.3)\\
&> \frac{(1 + \alpha_{\min})(n-1)}{(1 + \alpha_{\min})\frac{n-1}{n+1}} - n && (5.4)\\
&= 1\,. && (5.5)
\end{aligned}$$

Thus, for every x such that $0 \le \mathcal{S}(x) < 1 + \epsilon$, we have $\Delta(x) > 1$.

Step II: *For every time t there exists a time $t' > t$ such that $\mathcal{S}(x^{t'}) \ge 1 + \epsilon$.*

By definition $\mathcal{S}(x^t) \ge 0$ for every x^t. In step I we show that $\Delta(x) > 1$ when $0 \le \mathcal{S}(x) < 1 + \epsilon$. Now, combining with the fact that the learning rate is such that $\sum_{\tau=t'}^{t} \eta_\tau \to \infty$ as $t \to \infty$, the claim follows.

Step III: *There exists a time $\bar{t}$ such that $\eta_t \Delta(x^t) > -\epsilon$ for every $t > \bar{t}$.*

Notice that $\Delta(x^t)$ is bounded from below:

$$\Delta(x^t) = \sum_i \left((1 + \alpha_i) \frac{\mathcal{S}(x^t) - x_i^t}{(\mathcal{S}(x^t))^2} - \alpha_i \right) > -\sum_i \alpha_i \ge -n,$$

where the last inequality follows our assumption that $\alpha_i \le 1$, for every $i \in N$. By our assumption on the learning rate there exists at time $\bar{t}$ such that $\eta_{\bar{t}} \leq \epsilon/n$, and the claim follows.

Step IV: *If $t_1 > \bar{t}$ and $\mathcal{S}(x^{t_1}) > 1$ then $\mathcal{S}(x^t) > 1$ for every $t > t_1$.*

The proof is by induction on t. For the induction base $t = t_1$, the claim holds trivially. Assume the induction hypothesis holds for some $t > t_1$. If $\mathcal{S}(x^t) \in (1, 1+\epsilon]$ then $\Delta(x^t) > 0$ and

$$\mathcal{S}(x^{t+1}) = \mathcal{S}(x^t) + \Delta(x^t)\eta_t \geq \mathcal{S}(x^t) > 1,$$

Otherwise, $\mathcal{S}(x^t) > 1 + \epsilon$, but then $\eta_t \Delta(x^t) > -\epsilon$ and

$$\mathcal{S}(x^{t+1}) = \mathcal{S}(x^t) + \Delta(x^t)\eta_t \geq \mathcal{S}(x^t) - \epsilon > 1.$$

and the claim follows.

Step V: *There exists a time t^* such that $\mathcal{S}(x^t) > 1$ for every $t > t^*$.*

It follows from step III that after time $\bar{t}$, $\Delta(x^t) > -\epsilon$ for every $t > \bar{t}$. From step II it follows that there exists a time $t^* > \bar{t}$ such that $\mathcal{S}(x^{t^*}) \geq 1 + \epsilon$. From step IV, it follows that for every $t > t^*$, $\mathcal{S}(x^t) > 1$.

Step VI: *If $\eta_t = 1/\sqrt{t}$ then there exists $t^* = O(n^4)$, such that $\mathcal{S}(x^t) > 1$ for every $t > t^*$.*

If $\eta_t = \frac{1}{\sqrt{t}}$, then $\eta_t \Delta(x^t) > -\epsilon$ for every $t > \bar{t}$ where $\bar{t} = (n/\epsilon)^2$, and

$$\begin{aligned}
\frac{n}{\epsilon} &= \frac{n}{\frac{n-1}{n+1}\alpha_{\min} - \frac{2}{n+1}} && (5.6)\\
&\leq \frac{n}{\frac{3}{n+1} - \frac{2}{n+1}} && (5.7)\\
&= n(n+1) && (5.8)\\
&< 2n^2 && (5.9)
\end{aligned}$$

Hence, $\bar{t} < 4n^4$. Note that if $\alpha_{\min} = \Omega(1)$ then $\epsilon = \Omega(1)$ and $\bar{t} = O(n^2)$.

Now, if $\mathcal{S}(x^{\bar{t}}) > 1$, then $\mathcal{S}(x^t) > 1$ for every $t > \bar{t}$, as it follows from step IV. If not, then set $t^* = \bar{t} + 4n^2$. If $\mathcal{S}(x^t) > 1$ for some $\bar{t} < t \leq t^*$ then we are done. Otherwise, we have that $\Delta(\mathcal{S}(x^t)) = n$ for every such t. In this case

$$\mathcal{S}(x^{t^*}) = \mathcal{S}(x^{\bar{t}}) + \sum_{t=\bar{t}}^{t^*-1} \frac{1}{\sqrt{t}}\Delta(x^t) \geq 0 + 4n^2 \frac{1}{\sqrt{4n^4+4n^2}} n > 2,$$

since $n \geq 4$. But, this is in contradiction to the assumption that $\mathcal{S}(x^t) \leq 1$ for $\bar{t} < t \leq t^*$. Thus, at some time $t' < t^* = O(n^4)$, $\mathcal{S}(x^{t'}) > 1$ and therefore, following Step IV, for every $t > t^*$, $\mathcal{S}(x^t) > 1$. ∎

Proof of Theorem 5.4. By Lemma 5.5 there exists time t^*, such that $\mathcal{S}(x^t) > 1$, for every $t > t^*$. Using Lemma 5.3, we obtain that from t^* on, the agents are a posteriori playing the socially concave game $\Gamma^{\mathrm{TD}'}$.

The regret of agent i at a time $t > t^*$ is at most

$$\mathcal{R}_i(t) = O(\sqrt{t-t^*}) + O(t^*) = O(t^* + \sqrt{t}),$$

where the first term is the regret accumulated after t^* and $O(t^*)$ is an upper bound on the difference between the utility from the best fixed transmission rate (+1) and the worst possible loss (-1).

Thus, following Theorem 4.3, we conclude that at time $t > t^*$, the average strategy profile is an $O(n/\sqrt{t})$-Nash equilibrium, and that the average utility of each agent differs by at most $O(n/\sqrt{t})$ from her utility in that ϵ-Nash equilibrium. ∎

5.3 RED Policy

In random early discard the router drops packets as a function of the queue size. Since we do not have a queue size in our models, we model this by dropping fraction of the rate at each time step as a function of the offered load $\mathcal{S}(x)$.

Assume that the router drops packets at a rate of $\beta\mathcal{S}(x)$. This implies that user i would have an effective bandwidth of $x_i(1-\beta\mathcal{S}(x))$, and the total effective bandwidth is $\mathcal{S}(x)(1-\beta\mathcal{S}(x))$. Since the capacity of the link is $C = 1$ we need that $\mathcal{S}(x)(1-\beta\mathcal{S}(x)) \leq 1$, which always holds for $\beta \geq 1/4$ (also, we can not dropping more than $\mathcal{S}(x)$ units of rate, so $\beta \leq 1$).

In this case, for $\beta \geq 1/4$ we derive the RED utility function,

$$u_i^{\text{RED}}(x) = \begin{cases} x_i(1 - \beta\mathcal{S}(x)) - \alpha_i\beta x_i\mathcal{S}(x) & \mathcal{S}(x) \leq 1/\beta \\ -\alpha_i x_i & \mathcal{S}(x) > 1/\beta \end{cases} \tag{5.10}$$

Notice that $u_i^{\text{RED}}(1/\beta, x_{-i}) > u_i^{\text{RED}}(x_i, x_{-i})$, for every $x_i > \frac{1}{\beta}$, $x_{-i} \in \mathbb{R}_+^{n-1}$, since $\frac{\partial}{\partial x_i}u^{\text{RED}}(x) < 0$ for every such x. We therefore set an agent's strategy space to $[0, \frac{1}{\beta}]$. The game associated with the RED policy will be denoted by $\Gamma^{\text{RED}} = (N, ([0, \frac{1}{\beta}])_{i \in N}, (u_i^{\text{RED}})_{i \in N}])$

Theorem 5.6. *If every agent in a Γ^{RED} game plays according to the GIGA procedure, then the average strategy vector will converge to a Nash equilibrium, and the average utility of each agent will converge to her utility at that Nash equilibrium.*

(i) The average profile of actions will be an ϵ-Nash equilibrium, where $\epsilon = O\left(\frac{1}{\sqrt{t}}\right)$

(ii) The average utility of each agent will differ from her utility at that Nash equilibrium by at most $O\left(\frac{1}{\sqrt{t}}\right)$.

Furthermore, if every agent runs GIGA with $\eta_t = 1/\sqrt{t}$, then at every time $t > t^*$, where $t^* = O\left(\left(\frac{n}{\alpha_{\min}}\right)^2\right)$ and $\alpha_{\min} = \min_{i \in N} \alpha_i$,

The game Γ^{RED} is not necessarily a socially concave game (*e.g.*, the utility of an agent is generally not concave). However, the utility function in Γ^{RED} could be modified as follows to get a socially concave game. We define a new game denoted $\Gamma^{\text{RED}'} = (N, ([0, 1])_{i \in N}, (q_i^{\text{RED}})_{i \in N})$, where

$$q_i^{\text{RED}}(x) = x_i(1 - \beta\mathcal{S}(x)) - \alpha_i\beta x_i\mathcal{S}(x).$$

Lemma 5.7. *The game $\Gamma^{\text{RED}'}$ is a socially concave game.*

Proof. Fix x_i, and consider the function

$$f(z) = x_i(1 - \beta(x_i + z)) - \alpha_i\beta x_i(x_i + z).$$

The function f is linear in z and consequently, $q_i^{\text{RED}}(x)$ is linear in x_{-i}, as a composition of two linear functions.

Now, consider the function g defined as the sum of all agents utilities,

$$g(x) = \sum_{i \in N} q_i^{\text{RED}}(x) = \sum_{i \in N} x_i - \sum_{i \in N} \beta(1+\alpha_i)x_i\mathcal{S}(x) = \mathcal{S}(x) - S^2(x)\sum_{i \in N}\beta(1+\alpha_i).$$

The function g can be composed as $g(x) = h(\mathcal{S}(x))$, where $h(z) = z - z^2\sum_{i\in N}\beta(1+\alpha_i)$. Hence, g is a concave function as composition of a concave function with a linear function.

Thus, $\Gamma^{\text{RED}'}$ is socially concave, as both properties (A1) and (A2) of Definition 4.1 hold.

∎

Lemma 5.8. *If every agent in a Γ^{RED} game, plays according to the GIGA procedure, then there exists a time t^* such that for every $t > t^*$, $\mathcal{S}(x^t) < 1/\beta$. Furthermore, if the learning rate $\eta_t = \frac{1}{\sqrt{t}}$, then $t^* = O((\frac{n}{\alpha_{min}})^2)$.*

Proof. Let $\Delta_i^t(x) = \frac{\partial}{\partial x_i}u_i^{\text{RED}}(x^{t-1})$ and let $\Delta(x) = \sum_{i\in N}\Delta_i(x)$. The proof proceeds in a number of steps.

Step I: *If x is such that $\frac{1}{\beta(\alpha_{\min}+1)} \leq \mathcal{S}(x) < \frac{1}{\beta}$, then $\Delta(x) \leq -1$.*

If $\mathcal{S}(x) \leq 1/\beta$ then

$$\frac{\partial}{\partial x_i}u_i^{\text{RED}}(x^t) = 1 - (\beta + \alpha_i\beta)\sum_{j\neq i}x_j - 2(\beta+\alpha_i\beta)x_i = 1 - (\beta+\alpha_i\beta)(\mathcal{S}(x) + x_i)$$

and

$$\begin{aligned}
\Delta(x) &= n - \sum_{i\in N}(\beta+\alpha_i\beta)(\mathcal{S}(x)+x_i) && (5.11)\\
&\leq n - (n+1)\beta(1+\alpha_{\min})\mathcal{S}(x) && (5.12)\\
&\leq n - (n+1)\beta(1+\alpha_{\min})\frac{1}{\beta(\alpha_{\min}+1)} && (5.13)\\
&= -1. && (5.14)
\end{aligned}$$

Step II: *For every time t there exists a time $t' > t$ such that $\mathcal{S}(x^{t'}) \leq 1/\beta$.*

For x such that $\mathcal{S}(x) > 1/\beta$, we have $\frac{\partial}{\partial x_i}u_i^{\text{RED}}(x^t) = -\alpha_i$, and $\Delta(x) = \sum_{i\in N} -\alpha_i < 0$. Combining with the fact that the learning rate is such that $\sum_{\tau=t'}^{t}\eta_\tau \to \infty$ as $t\to\infty$, the claim follows.

Step III: *There exists a time $\bar{t}$ such that $\eta_t \Delta(x^t) < \epsilon$ for every $t > \bar{t}$, where $\epsilon = \frac{1}{\beta} - \frac{1}{\beta(\alpha_{\min}+1)}$.*

If x^t is such that $\mathcal{S}(x^t) > \frac{1}{\beta}$ then $\Delta(x^t) < 0$, as we showed in Step II. Otherwise, as observed clearly in equation (5.11), $\Delta(x^t) < n$. By our assumption on the learning rate there exists at time $\bar{t}$ such that $\eta_{\bar{t}} \leq \epsilon/n$, and the claim follows.

Step IV: *If $t_1 > \bar{t}$ and $\mathcal{S}(x^{t_1}) < 1/\beta$ then $\mathcal{S}(x^t) < 1/\beta$ for every $t > t_1$.*

The proof is by induction on t. For the induction base $t = t_1$, the claim holds trivially. Assume the induction hypothesis holds for some $t > t_1$. If $\mathcal{S}(x^t) \in [\frac{1}{\beta(1+\alpha_{\min})}, \frac{1}{\beta}]$ then $\Delta(x^t) < -1$ and

$$\mathcal{S}(x^{t+1}) = \mathcal{S}(x^t) + \Delta(x^t)\eta_t \leq \mathcal{S}(x^t) < \frac{1}{\beta},$$

Otherwise, $\mathcal{S}(x^t) < \frac{1}{\beta(1+\alpha_{\min})}$, but then $\eta_t \Delta(x^t) < \epsilon$ and

$$\mathcal{S}(x^{t+1}) = \mathcal{S}(x^t) + \Delta(x^t)\eta_t < \frac{1}{\beta(1+\alpha_{\min})} + \left(\frac{1}{\beta} - \frac{1}{\beta(1+\alpha_{\min})}\right) = \frac{1}{\beta}.$$

and the claim follows.

Step V: *There exists a time t^* such that $\mathcal{S}(x^t) > 1$ for every $t > t^*$.*

It follows from step III that $\Delta(x^t) < \epsilon$, for every $t > \bar{t}$. From step II it follows that there exists a time $t^* > \bar{t}$ such that $\mathcal{S}(x^{t^*}) < \frac{1}{\beta}$. And, from step IV, it follows that for every $t > t^*$, $\mathcal{S}(x^t) < \frac{1}{\beta}$. This completes the proof of the first part of the claim.

Step VI: *If $\eta_t = 1/\sqrt{t}$ then there exists $t^* = O(n^2(\frac{1}{\alpha_{\min}})^2)$, such that $\mathcal{S}(x^t) > 1$ for every $t > t^*$.*

Since $\eta_t = \frac{1}{\sqrt{t}}$, then $\eta_t \Delta(x^t) < \epsilon$ for every $t > \bar{t}$ where

$$\begin{aligned}
\bar{t} &= (n/\epsilon)^2 && (5.15)\\
&= \left(\frac{n}{\frac{1}{\beta} - \frac{1}{\beta(1+\alpha_{\min})}}\right)^2 && (5.16)\\
&= \left(\frac{n}{\alpha_{\min}}\beta(1+\alpha_{\min})\right)^2 && (5.17)
\end{aligned}$$

Now, if $\mathcal{S}(x^{\bar{t}}) < 1/\beta$, then $\mathcal{S}(x^t) < 1/\beta$ for every $t > \bar{t}$, as it follows from step IV. If not, then set $t^* = 64(n/\epsilon)^2 + 1$. If $\mathcal{S}(x^t) < 1/\beta$ for some $\bar{t} < t \leq t^*$ then we are done. Otherwise, we have that $\Delta(\mathcal{S}(x^t)) < -n \cdot \alpha_{\min}$ for every such t. Recall that no agent ever transmits with a rate greater than $\frac{1}{\beta}$, hence at time $\bar{t}$, $\mathcal{S}(x^{\bar{t}}) \leq \frac{n}{\beta} \leq 4n$. Therefore,

$$
\begin{aligned}
\mathcal{S}(x^{t^*}) &= \mathcal{S}(x^{\bar{t}}) + \sum_{t=\bar{t}}^{t^*-1} \frac{1}{\sqrt{t}} \Delta(x^t) && (5.18)\\
&\leq 4n + (\sum_{t=\bar{t}}^{t^*-1} \frac{1}{\sqrt{t}})(-n\alpha_{\min}) && (5.19)\\
&\leq 4n + (\sum_{t=\bar{t}}^{t^*-1} \frac{1}{\sqrt{t^*-1}})(-n\alpha_{\min}) && (5.20)\\
&= 4n - (t^* - \bar{t})(n\alpha_{\min}) \frac{1}{\sqrt{t^*}} && (5.21)\\
&= 4n - 63(n/\epsilon)^2 (n\alpha_{\min}) \frac{1}{8(n/\epsilon)} && (5.22)\\
&= 4n - \frac{63}{8} n^2 (1 + \alpha_{\min})\beta && (5.23)\\
&= 4n - \frac{63}{32} n^2 (1 + \alpha_{\min}) && (5.24)\\
&\overset{(a)}{\leq} \frac{1}{8} && (5.25)\\
&< \frac{1}{\beta}, && (5.26)
\end{aligned}
$$

where (a) holds for every $n \geq 2$, which is our case. But, this is in contradiction to the assumption that $\mathcal{S}(x^t) \geq \frac{1}{\beta}$ for every $\bar{t} < t \leq t^*$. Thus, at some time t, $\bar{t} < t \leq t^*$, $\mathcal{S}(x^t) \leq \frac{1}{\beta}$ and therefore, following Step IV, for every $t > t^*$, $\mathcal{S}(x^t) > 1$. This completes the proof of the second part of the claim. ∎

Proof of Theorem 5.6. From Lemma 5.8 we learn that after some time t^*, the profile of actions is such that $\mathcal{S}(x^t) \leq \frac{1}{\beta}$, for every $t > t^*$. It follows Lemma 5.7, that from t^* on, the agents are a posteriori playing the socially concave game, $\Gamma^{\mathrm{TD}'}$.

The regret of agent i at a time $t > t^*$ is at most

$$\mathcal{R}_i(t) = O(\sqrt{t - t^*}) + O(t^*) = O(t^* + \sqrt{t}),$$

where the first term is the regret accumulated after t^* and $O(t^*)$ is an upper bound on the difference between the utility from the best fixed transmission rate (+1) and the worst possible loss (-1).

Thus, following Theorem 4.3, we conclude that at time $t > t^*$, the average strategy profile is an $O(n/\sqrt{t})$-Nash equilibrium, and that the average utility of each agent differs by at most $O(n/\sqrt{t})$ from her utility in such an equilibrium. ∎

Chapter 6

Strategic Protocols for Queue Management

In this chapter we deal with latency issues arising in the context of network devices such as switches and routers. Given multiple streams of incoming packets which are to be merged and sent along the same outgoing link, limited bandwidth on the outgoing link may result in packet delay and/or packet loss.

We consider the online problem of active queue management. In our problem the input consists of an online sequence of packets arriving over time to a non-preemptive FIFO queue. Any packet placed in the queue will eventually be sent, and packets are sent in order of arrival.

We consider queuing models where the actual benefit derived from sending a packet degrades over time. The goal is to model transmission of packets bearing time-dependent data such as audio, video, and more general latency sensitive applications. The latency of a packet is a major parameter in the assessment of its true utility (*e.g.*, TCP, where delayed messages may result in a reduction of the overall transmission protocol throughput).

In the bounded delay model [69] incoming packets have step function describing the value loss, past the deadline the packet is useless, prior to the deadline it has lost no value. Unfortunately, online queue policies for FIFO queues and heterogenous packets with deadlines result in unbounded competitive ratios. Quite likely, this is why only non-FIFO queue regimes have been studied for service with deadlines.

In this chapter we show that online policies for FIFO queues are much better if one assumes more gradual value loss, (constant competitive ratio *vs.* unbounded ratios). So, the question arises, which is the more relevant model? One obvious reason in support of the gradual loss of value lies within human nature — people are generally not entirely oblivious to the time "wasted" while waiting for service, even if such service is eventually given.

Consider the example of streaming video, there is an inherent deadline imposed by the timestamp within an MPEG video packet. However, such packets will typically proceed through a large network with multiple routers along the transmission path. Thus, although there is a true deadline, this deadline is only relevant with regard to the entire transmission path taken, not in the context of any specific router. Furthermore, taking no value loss prior to the deadline may imply that the packet will indeed pass through (some) of the routers, but will be discarded further down the line. A reasonable rule of thumb could be to assume that value is lost over time, and this may produce much better overall performance.

We consider several versions of this problem. The simplest case is that of homogenous packets, where all packets have the same intrinsic value upon arrival, and packets lose value linearly over time. Our results also hold for "growing impatience" rather than linear value loss, *i.e.*, the longer the packet is in the system, the more value is lost per time unit. In particular, this includes commonly considered "soft real time restrictions" such as quadratic or exponential value loss (in terms of delay).

We next deal with the more general case of heterogenous packets, where different packets may have different intrinsic values upon arrival, bounding the competitive ratio to be some constant $4.23 \leq c < 8$.

Last, we consider a much more general model where packets not only differ in their value for service, but also in the way they value time. This model also generalized the bounded delay model, and thus, a FIFO policy would yield an unbounded competitive ratio. To come up with positive results, we give more power to the queue manager, and allow non-preemptive queue management, and discard the FIFO requirement.

Our results include:

1. For homogenous packets (equal valued) we give a lower bound of ϕ on the competitive ratio (even for randomized algorithms).

2. For heterogenous packets and linear value loss:

    (a) We give a simple threshold queue policy, "doubling threshold", with a competitive ratio of at most 8.

(b) We observe that this problem has an $O(n \log n)$ time optimal offline algorithm (improving upon the obvious matching based approach).

(c) We show a lower bound of 4.23 on the competitive ratio of any deterministic online algorithm.

3. For heterogeneity of time loss factors, we show that the greedy algorithm attains a competitive ratio of 2, in a very general time-loss model. For the special case of heterogenous value, and heterogenous but constant loss per a unit of delay, we give a randomized algorithm with a competitive ratio $e/(e-1) \approx 1.58$.

4. Finally, we relate the issue of the online competitive ratio to an online mechanism design problem for packets generated by selfish agents. In this case there is little reason to trust the "intrinsic value" claimed by the owner. We reinterpret our online algorithms as yielding an incentive compatible online pricing scheme for heterogenous packets, that guarantees a constant fraction of the optimal social welfare (defined as the sum of agent utilities).

6.1 Preliminaries

We consider a single, non-preemptive, queue of packets. Time is partitioned into unit length time slots, integrally aligned. Packets arrive over time at non-integral times. Multiple packets may arrive during a single time slot, but every packet has its own unique arrival time. At the end of *every* time slot, and should the queue be non-empty, one (and exactly one) packet is extracted from the queue and transmitted.

A packet p is identified by its value, denoted $\mathrm{val}(p)$ and its (distinct, non-integer) time of arrival, denoted $\mathrm{arrive}(p)$. A packet is either rejected or inserted into the queue. A packet placed in the queue must eventually be transmitted, at the end of some time slot, which we call $\mathrm{send}(p)$. The delay a packet experiences is the number of full time slots between arrival and transmission times of the packet, *i.e.*, $\mathrm{delay}(p) = \lfloor \mathrm{send}(p) - \mathrm{arrive}(p) \rfloor$.

The delay of a packet is equal to the number of packets in the queue when it arrived (which does not include itself):

$$\begin{aligned}
\mathrm{delay}(p) &= \text{buffer size at time } \mathrm{arrive}(p) \\
&= \lfloor \mathrm{send}(p) - \mathrm{arrive}(p) \rfloor .
\end{aligned}$$

Expression	**Definition**
p	A single packet
σ	An event sequence; consists of arrival events, at arbitrary non integral times, and a send event at each time $t \in \{1, 2, \ldots\}$.
val(p)	The value of packet p.
arrive(p)	The time of packet p arrival.
send(p)	The time packet p is sent, in case it is sent.
delay(p)	The delay of packet p, rounded down delay$(p) = \lfloor \text{send}(p) - \text{arrive}(p) \rfloor$.
benefit(p)	The benefit of a packet p; If p is sent then benefit$(p) =$ val$(p) -$ delay(p), otherwise, benefit$(p) = 0$
ON	A threshold algorithm.
DT	Algorithm doubling thresholds.
$\pi(\sigma)$	The benefit of an online algorithm π on an events sequence σ.
$B(t)$	The buffer size at time t.
$\psi(t)$	A potential function.
R	The homogenous value.
ϕ	The golden ratio; $\phi \approx 1.618$;$\phi^3 \approx 4.23$.

Table 6.1: Notation in use.

In general, the loss of value of a packet is some function of the packet delay. In particular, linear loss means loss proportional to the delay, we will generally assume linear loss to mean a constant of one, but our results can be easily extended to other constants of proportionality. This means that the utility or benefit collected from sending a packet p, denoted benefit(p) is

$$\text{benefit}(p) = \text{val}(p) - \text{delay}(p).$$

The adversary determines the timing of packet arrivals and the value associated with the packet. Packets are transmitted from the head of the FIFO queue at integral times. We do not explicitly assume a restriction on the capacity of the queue. However, one may assume without loss of generality that no packet will be inserted into the queue if the delay is such that the loss is greater than the initial value.

An online queue policy specifies, for every incoming packet, as to whether to enqueue the packet or to reject it. For an online policy, the decision to accept or reject a packet may not depend on future events. An offline queue policy does the same but has prior knowledge of the entire event sequence in advance. The benefit of a queue policy on a given event sequence is the sum of the benefits of the packets that it transmits.

A memoryless queue policy determines whether to accept or reject an incoming packet

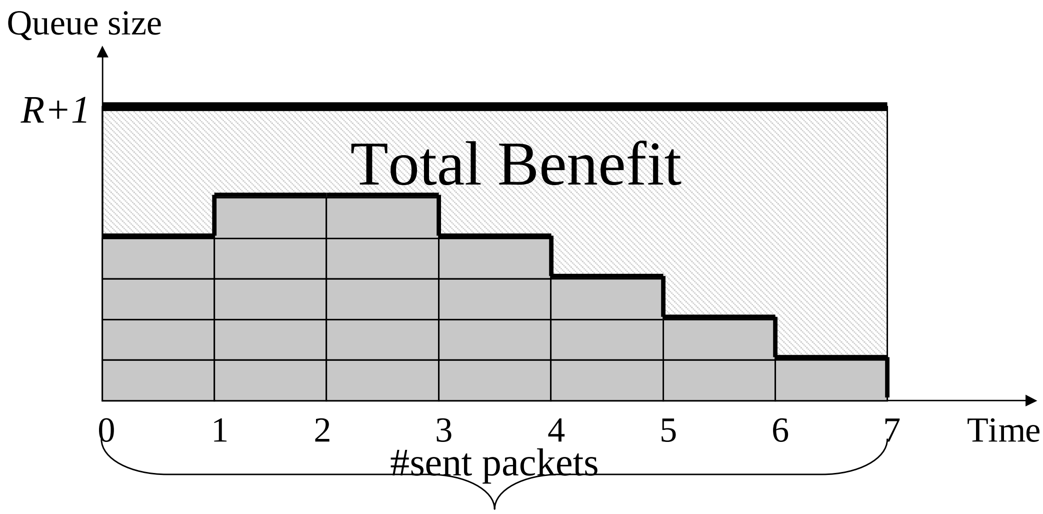


Figure 6.1: This figure depicts a snapshot of buffer sizes at send events, that is a snapshot of the algorithms buffer prior to send events at time $t = 1, 2, \ldots$. The area between the buffers' snapshot and the line $R + 1$, captures the overall benefit from this schedule.

based only upon the current contents of the buffer, the delay that these packets have had, and the value of the incoming packet. A special case of memoryless policies, called *threshold policies*, determine if to accept or reject while only considering the number of packets currently in the buffer, and the value of the incoming packet.

Let $B(t)$ denote the number of packets in the queue at time t. If a packet p arrives at time t, $B(t)$ is the number of packets in the queue, not counting p, (irrespective of whether p is to be accepted or not). Similarly, let $B^*(t)$ denote the number of packets in the queue of the optimal (offline) schedule.

As there may be several different optimal schedules (giving the maximal benefit), we fix a specific optimal policy, OPT, defined in Section 6.6.

6.2 Homogenous Packets

In this section we consider only sequences of homogenous packets, all having the same value $R > 0$. We begin with an observation regarding the relation between the sequence of buffers size at send events, for a particular schedule, and the benefit of that schedule.

Observation 6.1. *The benefit of a policy from a finite event sequence of homogenous* *packets with value R is*

$$\left|\{i \in \mathbb{N} | i \le t_{\max} \text{ and } B(i) > 0\}\right| \cdot R - \sum_{i \in \{1,\ldots,t_{\max}\}} (B(i) - 1), \tag{6.1}$$

where $t_{\max}$ is the transmission time of the last packet transmitted (i.e., the transmission time of the last packet accepted from the sequence).

Figure 6.1 a snapshot of the buffer state of some schedule are illustrated, together with the benefit of that schedule.

In Section 6.2.1 we present a deterministic algorithm (queue policy) with a competitive ratio of $\phi \approx 1.618$. In Section 6.2.2 we show that ϕ is a lower bound on the competitive ratio of all online algorithms, even for randomized algorithms against an oblivious adversary.

6.2.1 An optimal online algorithm

Our online policy for homogenous packets is a threshold policy.

As a motivating example, consider the following: Let B be the current size of the online buffer, consider the $2B$-threshold policy, in which a packet is accepted if its value is greater or equal to twice the delay *i.e.*, its benefit is at least $2B$. Alternately, as all packets have value R in the homogenous case, accept a packet only if the current size of the buffer is no more than $\lfloor R/2 \rfloor$. It can be argued that the competitive ratio is close to 4. A $2B$-threshold policy accepts about half the number of packets accepted by an optimal policy (no optimal policy would ever have more than R packets in its queue, otherwise packets with 0 or negative benefit would be queued). For every packet accepted, the $2B$-threshold policy collects a benefit of at least $\lceil R/2 \rceil$, which is at least half the maximum benefit collected by OPT from any packet. We next show a more refined analysis of the competitive ratio.

Theorem 6.2. *Let B be the current size of the queue. The competitive ratio of the $(B\phi^2)$-threshold algorithm is at most $\phi + \epsilon(R)$, where $\epsilon(R) = O(1/R)$.*

In our proof we use ON to denote the $(B\phi^2)$-threshold policy while OPT denotes the optimal offline. We first perform a simple relaxation on the event sequence, defined as follows: After ON's buffer is emptied, there are no more packet arrival events until OPT's buffer is emptied as well. We derive the following claim for such restricted event sequences:

Lemma 6.3. *For any queue policy, limiting the event sequences to be restricted event sequences does not decrease the competitive ratio.*

Proof. We consider an arbitrary event sequence σ which does not meet the "restricted" criteria, *i.e.*, there is some time t_0 at which ON's queue is empty and the next packet arrival is at time t_1 such that the OPT queue is non-empty, $B^*(t_1) > 0$.

We change event sequence σ to σ' by adding $B^*(t_0)$ send events following time t_0, and delaying all other events in σ by an extra $B^*(t_0)$ time units. The purpose of the additional send events is to empty OPT's buffer. We show that the ratio between OPT's benefit and the benefit for ON on σ is no smaller than the same ratio on σ'.

ON's decisions remain unchanged, since the additional send events are scheduled when ON's queue is empty. On the other hand, the optimal offline benefit cannot decrease. Keeping the same schedule produces a benefit of at least OPT(σ), as the queue size at every send event does not increase comparing to the original. Therefore OPT(σ') $\geq$ OPT(σ) and the claim follows. ∎

Once both ON and OPT have empty buffers, they are both back at their initial state. Thus, it suffices to prove an upper bound on the competitive ratio by restricting attention to those restricted event sequences that have no arriving packets after ON empties its buffer for the first time. We assume that the sequence begins with at least one arrive event before the first send event. In the remainder of this section we restrict attention to such event sequences.

Thus, for our event sequences, the online algorithm sends one packet at every send event until its queue is empty.

We now prove Theorem 6.2. We define a potential function that measures the difference between twice the benefit of ON, and the benefit of OPT. We show inductively that the potential function is always non-negative.

We define the functions $f : R^+ \mapsto Z^+$ and $v : R^+ \mapsto Z^+$ as follows:

$$\begin{aligned} f(t) &= \left|\{i \in \mathbb{N} | i < t \text{ and } B(i) > 0\}\right| \cdot R - \sum_{i \in \{1,\ldots,\lfloor t \rfloor\}} (B(i) - 1), \\ v(t) &= \sum_{j=1}^{B(t)} R - (j-1). \end{aligned}$$

Consider an event sequence where the last packet arrival event is at time t_{last}, it follows from Observation 6.1 and the definition of f and v above that for any $t > t_{\text{last}}$ the total benefit of the queue policy on the event sequence is equal to $f(t)+v(t)$. Similarly, we define the functions $f^*(\cdot)$, and $v^*(\cdot)$, where the function $B^*(\cdot)$ replaces the function $B(\cdot)$. It then follows that when both ON and OPT buffers are empty then $f(t)$ and $f^*(t)$ are respectively equal to the benefit of ON and OPT on the event sequence, and both $v(\cdot)$ and $v^*(\cdot)$ are zero.

Claim 6.4. *If $B(t) > 0$ then $\phi \cdot f(t) \geq f^*(t)$.*

Proof. Obviously $f^*(t) \leq \lfloor t \rfloor \cdot R$. If $B(t) > 0$ then for every $t' < t$ we have $B(t') > 0$, since the online buffer empties only once. Hence, $\left|\{i \in \mathbb{N} | t > i \text{ and } B(i) > 0\}\right| = \lfloor t \rfloor$. Also, ON's queue length never exceeds $(1 - 1/\phi)R$. Therefore,

$$f(t) \geq \lfloor t \rfloor \cdot R - \lfloor t \rfloor (1 - 1/\phi)R \geq 1/\phi \cdot f^*(t) \ .$$

∎

We now define a potential function $\psi : R^+ \mapsto Z$ as follows:

$$\psi(t) = (\phi + \epsilon(R))\,(f(t) + v(t)) - (f^*(t) + v^*(t))\,.$$

Proving that $\psi(t) \geq 0$ for all $t \geq 0$ concludes the proof of Theorem 6.2: Note that the functions $f(\cdot), v(\cdot)$, and $B(\cdot)$ (and similarly, $f^*(\cdot), v^*(\cdot)$, and $B^*(\cdot)$), change only when events occur. We now use the notation that for the i'th event in the event sequence, occurring at time t, we write $f(i)$ to indicate the value of f just after the queue policy responded to the event (instead of $f(t+\epsilon)$, for an arbitrary small ϵ). (Similarly, we use notation $B(i), v(i)$, and $f^*(i)$, etc.).

Following Lemma 6.3 we consider only restricted sequences. We prove $\psi(t) \geq 0$ for all t by induction on the number of events. The basis of the induction is at time 0 before any event occurs. Both queues are empty and the claim trivially holds. Assume the claim holds for $i-1$ events, and consider the i'th event.

Assume the ith event is a send event. The value $f(i) = f(i-1) + R - (B(i-1) - 1)$. The queue size decreases by 1, hence, $v(B(i)) = v(B(i-1)) - (R - (B(i-1) - 1))$, and we get that $f(i) + v(i) = f(i-1) + v(i-1)$. Similarly, $f^*(i) + v^*(i) = f^*(i-1) + v^*(i-1)$, and therefore $\psi(i) = \psi(i-1)$.

Assume the ith event is packet arrival. The function f changes only after send events, hence $f(i) = f(i-1)$, and $f^*(i) = f^*(i-1)$. We consider now 4 relevant cases:

(i) Both OPT and ON accept the packet,

(ii) ON accepts and OPT rejects, and

(iii) ON rejects and OPT accepts.

(iv) A packet which both ON and OPT reject does not affect the potential function and can be ignored.

In case (i), $B(i) = B(i-1)+1$ and $B^*(i) = B^*(i-1)+1$. When a packet is accepted, the queue size after packet acceptance is at most $(1-1/\phi)R+1$, hence, $v(i) = v(i-1) + R - (B(i+1)-1) \geq v(i-1) + R/\phi$. OPT's queue size is at least 1 after the packet is accepted, so $v^*(i) = v^*(i-1) + R - (B^*(i+1)-1) \leq v^*(i) + R$. Consequently,

$$\begin{aligned} \psi(i) - \psi(i-1) &\geq \phi\left(v(i) - v(i-1)\right) - \left(v^*(i) - v^*(i-1)\right) \\ &\geq 0. \end{aligned}$$

In case (ii), ON's queue size increases by 1, hence, $v(i) > v(i-1)$, while OPT's queue size does not change, so $v^*(i) = v^*(i-1)$. Again, $\psi(i) > \psi(i-1)$.

In case (iii), in which a packet is rejected by ON must have occurred prior to the first time ON's queue is emptied and therefore, by Claim 6.4

$$\phi f(i) \geq f^*(i). \tag{6.2}$$

Additionally, we can tell that $B(i) = B(i-1) = \lfloor (1-1/\phi)R \rfloor + 1$ since ON just rejected a packet. OPT's queue size $B^*(i) \leq R$. Thus,

$$\begin{aligned} v(i) &= R + (R-1) + \cdots + (R - \lfloor (1-1/\phi)R \rfloor) \\ &= 1/2(\lfloor \frac{R}{\phi^2} \rfloor + 1)(2R - \lfloor \frac{R}{\phi^2} \rfloor) \\ &\geq 1/2(\frac{R}{\phi^2} - 1 + 1)(2R - R/\phi^2) \\ &= \frac{R^2}{2\phi} \end{aligned}$$

On the other hand, $v^*(i) \leq R + (R-1) + \cdots + (R - \lfloor R \rfloor) \leq 1/2(R+1)^2$.

Setting $\epsilon(R) = \phi(2/R + 1/R^2)$ we get that

$$(\phi + \epsilon(R))v(i) \;\geq\; 1/2(R+1)^2 \;\geq\; v^*(i). \tag{6.3}$$

Combining equations (6.2), and (6.3) we get that $\psi(i) \geq \psi(i-1)$. We conclude that $\psi(i) \geq 0$ after every event i. This completes the proof of Theorem 6.2.

6.2.2 Lower bound

The following theorem shows that any online algorithm for homogenous packets has a competitive ratio of at least $\phi \approx 1.618$ and thus the $(B\phi^2)$-threshold algorithm achieves the best competitive ratio possible.

Theorem 6.5. *The competitive ratio of any online algorithm (**deterministic** or **randomized**) is at least $\phi \approx 1.618$.*

Proof. We derive the proof for a deterministic online algorithm and at the end sketch the extensions for the randomized case.

Let π be an online policy for queue management. We construct an input sequence $\sigma(\pi)$ in the following way. We first set a threshold at αR where $\alpha = 1/\phi^2$. At each time slot R packets arrive until the first send event after which π's queue size is $(1/\phi^2)R = \alpha R$ or below. Subsequently, no further packets arrive. Let t_0 denote the time of the send event at which this occurs.[1]

The optimal schedule accepts one packet in every slot before slot number t_0. In the last slot the optimal schedule accepts all R packets. The optimal benefit for this sequence is therefore:

$$\mathrm{OPT}(\sigma(\pi)) = t_0 \cdot R + R(R+1)/2 \;.$$

We give an upper bound on the benefit of any policy π on event sequence $\sigma(\pi)$ using

[1]Notice that t_0 may be infinite if π's queue size never gets below αR. In such a case we can take t_0 to be arbitrarily large, and a similar argument would derive the lower bound.

Observation 6.1,

$$\begin{aligned}
\pi(\sigma(\pi)) &\overset{(a)}{\leq} t_0R - t_0\alpha R + \alpha R\cdot R - \frac{\alpha R}{2}(\alpha R - 1) \\
&= t_0\cdot R(1-\alpha) + \frac{\alpha R}{2}(2R - \alpha R + 1) \\
&< t_0\cdot R(1-\alpha) + \frac{\alpha R}{2}(2(R+1) - \alpha(R+1)) \\
&= (1-\alpha)t_0R + \alpha(2-\alpha)R(R+1)/2 ,
\end{aligned}$$

where (a) follows from Observation 6.1.

Since $\alpha = 1/\phi^2$, we have $1-\alpha = \alpha(2-\alpha)$ and so,

$$\begin{aligned}
\pi(\sigma(\pi)) &\leq (1-\alpha)\left(t_0R + R(R+1)/2\right) \\
&= (1-\alpha)\mathrm{OPT}(\sigma(\pi)).
\end{aligned}$$

The competitive ratio of every deterministic online algorithm is therefore at least $1/(1-\alpha) = \phi$.

For an online randomized algorithm we use a similar construction, however now the value $B(t)$ is a random variable. We now define t_0 to be the first time t when $\mathrm{E}[B(t)]$ is less than αR. The analysis of the optimum remains the same, while for the analysis of the randomized online we have a dependence on various values of $B(t)$.

In each time slot earlier than t_0 the expected queue size is at least αR and therefore the expected benefit is at most $(1-\alpha)R$.

$$\begin{aligned}
\mathrm{E}[\pi(\sigma(\pi))] &\leq t_0R - t_0\alpha R + \mathrm{E}[B(t_0)\cdot R - \frac{B(t_0)}{2}(B(t_0)-1)] \\
&= t_0\cdot R(1-\alpha) + \mathrm{E}[B(t_0)]\cdot R - \frac{1}{2}(\mathrm{E}[B^2(t_0)] - \mathrm{E}[B(t_0)]) \\
&< t_0\cdot R(1-\alpha) + \mathrm{E}[B(t_0)]\cdot R - \frac{1}{2}(\mathrm{E}^2[B(t_0)] - \mathrm{E}[B(t_0)]) \\
&< t_0\cdot R(1-\alpha) + \alpha R\cdot R - \frac{1}{2}((\alpha R)^2 - \alpha R) \\
&< (1-\alpha)t_0R + \alpha(2-\alpha)R(R+1)/2.
\end{aligned}$$

The first inequality follows since $\mathrm{E}[B(t)] \geq \alpha R$ for $t < t_0$ and the second inequality follows since $\mathrm{E}[B^2(t_0)] > \mathrm{E}^2[B(t_0)]$. ∎

Combining Theorem 6.5, and Theorem 6.2, derives a tight bound on the competitive ratio of the $(1/\phi^2)R$-threshold algorithm.

Corollary 6.6. *The competitive ratio of the* $(1/\phi^2)R$*-threshold algorithm is exactly* ϕ*.*

6.3 Heterogenous Packets

In this section we consider event sequences where every packet may have a different intrinsic value. The first policy we present, *doubling threshold* (DT) is a dynamic threshold policy: when the queue size is B, packets are accepted if their value is at least $2B$. *I.e.*, if the packet benefit is at least the queue size at the time of arrival.

We first restrict the set of event sequences, defined as follows: If DT accepts a packet p, then we reduce $\mathrm{val}(p)$ to $2B(\mathrm{arrive}(p))$. *I.e.* the value of p is reduced to the lowest possible value for which p is still accepted by DT. We derive the following claim for such restricted inputs.

Lemma 6.7. *The competitive ratio attainable on restricted event sequences is equal to the competitive ratio on arbitrary event sequences.*

Proof. Consider any event sequence σ. We compose a modified sequence σ', where the value of every packet p which is accepted by DT, is reduced in σ' to $2B(\mathrm{arrive}(p))$. Algorithm DT admits the same set of packets in σ, and in σ'. The benefit of $\mathrm{OPT}(\sigma')$, is at least $\mathrm{OPT}(\sigma) + (\mathrm{DT}(\sigma) - \mathrm{DT}(sigma'))$, since the optimal schedule for σ would yield at least this benefit for σ'. Thus, the ratio between

$$\frac{\mathrm{OPT}(\sigma')}{\mathrm{DT}(\sigma')} \geq \frac{\mathrm{OPT}(\sigma) - (\mathrm{DT}(\sigma) - \mathrm{DT}(\sigma'))}{\mathrm{DT}(\sigma) - (\mathrm{DT}(\sigma) - \mathrm{DT}(\sigma')} \geq \frac{\mathrm{OPT}(\sigma)}{\mathrm{DT}(\sigma)},$$

where the last inequality follows since $\mathrm{OPT}(\sigma) \geq \mathrm{DT}(\sigma)$. ∎

Now, using Lemma 6.7 we show that the competitive ratio of algorithm DT is at most 8.

Theorem 6.8. *The competitive ratio of algorithm DT is at most 8.*

We regard each packet accepted by DT as two half packets. The idea behind this proof is to map every packet accepted by OPT to 'half' a packet accepted by DT. We use a greedy mapping rule: an incoming packet accepted by OPT is mapped to the first "available" half packet in DT's queue, according to the FIFO sending order. A half packet is available if and only if no packet enqueued by OPT is mapped to it, otherwise it is unavailable . If p is mapped to one of the half packets that comprise packet q, we say p is associated with q. We next show that a packet enqueued by OPT cannot be transmitted by OPT before the packet with which it is associated with is transmitted by DT.

Lemma 6.9. *Let p be a packet in OPT's queue and let q be a packet in DT's queue with which p is associated. Then* $\mathrm{send}(p) \geq \mathrm{send}(q)$.

Proof. We prove the claim by induction on the number of packets accepted by OPT. The first packet p in a sequence is accepted by DT (the size of the buffer is 0 at this time, so every packet with positive value must be accepted). OPT also accepts a packet during this slot (otherwise it is not optimal). The first packet accepted by OPT, q, is associated with p (it may be the same packet). Both p and q are first in a FIFO buffer and are scheduled to be transmitted at the next send event, therefore $\mathrm{send}(p) = \mathrm{send}(q)$.

Assume the induction hypothesis for $i-1$ packets and consider the i'th packet p, accepted by OPT. Let B^* denote the length of OPT's queue just prior to p being enqueued. By the induction hypothesis, there are at most B^* unavailable half packets in DT's queue, so there are at most $\lfloor B^*/2 \rfloor$ packets scheduled to be sent before the first available packet q. Hence, $\mathrm{send}(p) \geq \mathrm{send}(q)$. ∎

We now argue that this mapping process is well defined, *i.e.*, it cannot be that a packet enqueued in OPT's queue has no available packet in DT's queue — the mapping never runs a deficit of half packets.

Lemma 6.10. *Every packet enqueued by OPT has an available half packet in DT's queue at the time of its arrival.*

Proof. Consider a packet p accepted at time t by OPT. If p is accepted by both OPT and DT then both halves of p are available. Otherwise, p is accepted by OPT and rejected by DT. Its value is $\mathrm{val}(p) < 2B(t)$ and consequently OPT's queue size before accepting p, $B^*(t) < 2B(t)$, as buffer lengths are integral $B^*(t) \leq 2B(t) - 1$. Thus, of the $2B(t)$ half packets in the DT queue, at most $2B(t) - 1$ of them are unavailable and there remains at least one available half packet. ∎

Mapping each packet of OPT to a half of a packet in the DT queue is, in and of itself, insufficient to guarantee a constant competitive ratio. We could be associating two packets of large benefit with a packet of small benefit. To overcome this issue, part of the benefit from a packet p enqueued by DT is distributed to all packets currently in the DT queue at time arrive(p). Instead of benefit, we now consider "credit", where the credit of a packet is sum of all benefits distributed to it. Clearly, the total benefit equals the sum of all the credits given to packets.

More precisely, when a packet p is accepted by DT, and $\text{arrive}(p) = t$, then $\text{val}(p) = 2B(t)$, and $\text{benefit}(p) = B(t)$. We "redistribute" some of p's benefit as credit given to other packets. A packet p keeps only half its credit $(1/2 \cdot B(\text{arrive}(p)))$ and the rest is equally distributed among the $B(\text{arrive}(p))$ packets that are already waiting in the queue, so that each gets an extra $1/2$ unit of credit.

Lemma 6.11. *At a time t, the credit of a packet in DT's buffer, is at least $1/2 \cdot B(t)$.*

Proof. Consider a packet p in DT's FIFO queue at position i. When it entered, the queue size was at least i, so it has an initial credit of at least $1/2 \cdot i$. Also, half a credit point was given to p from each of the $B(t) - i$ packets on top of p queued afterwards. Overall, its credit is at least $1/2 \cdot B(t)$. ∎

To conclude the proof that the competitive ratio of DT is ≤ 8, consider a packet r, accepted by DT, and denote its credit by credit(r). At most two packets p, q are associated with r. Packet r credit at time arrive(p) is at least $B(\text{arrive}(p))/2$, while the value of p, $\text{val}(p) \leq 2B(\text{arrive}(p))$, by assumption on the event sequences. Likewise, packet r credit at time arrive(q) is at least $B(\text{arrive}(q))/2$ and $\text{val}(q) \leq 2B(\text{arrive}(q))$. Hence,

$$\begin{aligned} \text{credit}(r) &= \max\big(B(\text{arrive}(p))/2, B(\text{arrive}(q))/2\big) \\ &\geq (\text{val}(p) + \text{val}(q))/8. \end{aligned}$$

I.e., the credit of a packet is at least $1/8^{th}$ of the sum of values of the packets associated with it and so equal to at least $1/8^{th}$ of their benefit. Comparing the sum over packets accepted by DT and over packets accepted by OPT, completes the proof.

Next we derive a lower bound of 4.23 on the competitive ratio of any deterministic algorithm.

Lemma 6.12. *let β be some positive real constant, and define the sequence of real values $b_0, b_1, \ldots,$ as follows: $b_0 = 1$, and for all $i \geq 1$:*

$$b_i = \min\left\{ x \in \mathbb{R}^+ \mid \sum_{j=0}^{\lfloor x \rfloor}(x - j) = \beta \sum_{j=0}^{i-1}(b_j - j) \right\}. \tag{6.4}$$

Then, if there exists some ℓ such that $b_\ell - \ell < 0$ then no deterministic competitive algorithm can achieve a competitive ratio of $\leq \beta$.

Proof. Consider a sequence of packet arrivals, all of which arrive within the first time slot, and of values equal to the b_i sequence. The sequence has the property that any online algorithm *must* accept these packets so as to have a competitive ratio of less than β. If the packet valued b_j is not accepted then the adversary can pack the buffer with packets of this value. However, if $b_j - j < 0$ then the online algorithm is only losing value by accepting this b_j valued packet. ∎

Theorem 6.13. *The competitive ratio of any deterministic online algorithm for heterogenous packets is at least 4.23.*

Proof. To prove a lower bound of 4.23 on the competitive ratio of any algorithm for heterogenous packets, we make use of Lemma 6.12, and try to find the largest β for which there exists an ℓ such that $b_\ell - \ell < 0$ holds. Solving the recurrence with $\beta = 4.23$ gives $b_{1807} < 1807$. ∎

Remark 6.14. *In a subsequent work Feldman [38] shows analytically that a sequence $\{b_i\}_{0 \leq i}$ satisfying condition (6.4) in Theorem 6.12 a lower bound of β exists for every $\beta < \phi^3 \approx 4.23$.*

6.4 Preemptive Queue

In this section we consider a preemptive queue policy that enables the queue manager to discard packets after they have been accepted. Further more we do not require that the packets are sent according to their FIFO order; packets can be sent in arbitrary order. In addition we continue not to impose any limit on the buffer size.

In this section, in contrast to previous sections, an online scheduling algorithm simply selects at every send event the packet to be sent.

The fact that the online algorithm does not have to decide whether to accept or reject a packet immediately after its arrival, gives it additional strength. For the heterogeneous packets model we have discussed earlier, this is all that an online algorithm needs to guarantee a competitive ratio of 1. As we show in Section 6.6, the greedy algorithm that sends at every time step the packet with the current most value attains the optimal offline performance.

Allowing preemption and arbitrary sending order, allows us to handle much more difficult value model, where packet have time-loss heterogeneously. This means that the value lost for delaying one unit of time, can be much more general.

As in previous sections we assume that packets arrive online, and that the value gained by sending a packet p consists of an initial intrinsic value $\mathrm{val}(p)$, from which a penalty for the delay is subtracted, only now, the penalty may be any decreasing function $c : \mathbb{N} \to \mathbb{R}_+$ of $\mathrm{delay}(p)$. We denote this gain by $\mathrm{val}(p, t) = \mathrm{val}(p) - c(\mathrm{delay}(p))$ the value of a packet p if it is sent at time t. We assume that $\mathrm{val}_{p,t} \geq 0$ for all $t \in \mathbb{N}, t \geq \mathrm{arrive}(p)$. For ease of notation we set $v(p, t) = 0$ for all $t \in \mathbb{N}, t < \mathrm{arrive}(p)$, *i.e.*, a packet has no value prior to its arrival. We emphasis that although this change in notation, an online scheduling algorithm still has no knowledge regarding the future stream of incoming packets.

This packet value model generalizes both the bounded delay model [69], and the model discussed in Sections 6.1 to 6.3 . Notice that in the non-preemptive, FIFO model, the competitive ratio of every online scheduling algorithm is unbounded even in the heterogenous linear loss model.

The simplest algorithm for preemptive queue management is the *greedy* algorithm, that always sends the packet with the current largest benefit for service, *i.e.*, the packet that would be benefiting the most if it was to be sent at the next send event. We show that greedy already suffices to guarantee a competitive ratio of 2 for the most general value model with value time loss heterogeneity

Theorem 6.15. *The competitive ratio of* GREEDY is 2.

Proof. To show an upper bound of 2 on the competitive ratio of the greedy algorithm we use a primal-dual approach, which demonstrated as an effective tool for the analysis of online algorithms by Buchbinder and Naor [21].

We first write the offline problem of finding the optimal schedule as a linear program (LP). We denote by n the overall number of packet arrivals, and by m, the first time slot, at which the value of all packets reduces to 0. We introduce a variable $X_{i,t}$ for every packet $1 \le i \le n$, and every time slot $1 \le t \le m$, indicating whether packet i is scheduled to time t.

$$\begin{array}{lll} \text{(LP)}: & \text{Maximize} & \sum X_{i,t}\text{val}(i,t) & \\ & \text{subject to} & \sum_t X_{i,t} \le 1 & \forall i \in \{1,\dots n\}, \\ & & \sum_i X_{i,t} \le 1 & \forall t \in \{1,\dots m\}, \\ & & X_{i,t} \ge 0 & \forall i \in \{1,\dots n\}, t \in \{1,\dots m\}. \end{array} . \tag{6.5}$$

The dual of (LP), denoted (DUAL) is

$$\begin{array}{llll} \text{(DUAL)}: & \text{Minimize} & \sum_{i=1}^n Y_i + \sum_{t=1}^m Z_t & \\ & \textit{subject to} & Y_i + Z_t \ge \text{val}(i,t) & \forall i \in \{1,\dots n\}, t \in \{1,\dots m\} \\ & & Y_i \ge 0, & \forall i \in \{1,\dots n\} \\ & & Z_t \ge 0 & \forall t \in \{1,\dots m\} \end{array} . \tag{6.6}$$

We rephrase the greedy algorithm in terms of (LP), and (DUAL). We initially set all variables to be 0. At each time $t \in \{1, 2, \dots, m\}$ the greedy algorithm schedules the packet with the highest current value val(i, t), that was not previously scheduled *i.e.*,

$$i \in \text{argmax}_{j|X_{j,t'}=0\forall t' \le t}\{\text{val}(j,t)\}.$$

If i is not uniquely defined, then the algorithm sets i to an arbitrary element in the set $\text{argmax}_{j|X_{j,t'}=0\forall t'}\{\text{val}(j,t)\}$. The variable $X_{i,t}$ is then set to 1. This assignment is clearly a feasible solution of (LP). It also increases the objective function of (LP) by val(i, t). In addition, we assign val(i, t) to the dual variables Y_i and Z_t. Thus, the objective function of (DUAL) is at most twice of that of the (LP).

When the algorithm terminates, the assignment of the Z and Y variables make a feasible solution for (DUAL): for every $1 \le i \le n$, and $1 \le t \le m$, consider the constraint $Y_i + Z_t \ge w_{i,t}$. Either

(i) packet i is scheduled at time $t' \le t$, in this case $Y_i = \text{val}(i, t') \ge \text{val}(i, t)$ or,

(ii) packet i is not scheduled on or before time t, in this case, if val$(i, t) = 0$, then the constraint is trivially satisfied. Otherwise, by definition of the greedy algorithm, there must be a different packet j which is scheduled time t, with value $\text{val}(j, t) \ge \text{val}(i, t)$. In this case, $Z_t = \text{val}(j, t)$, and the constraint is again satisfied.

From the weak duality theorem we know that any feasible value of the dual program serves as an upper bound on the value of the primal program. It follows then, that the assignment x is a 2-approximation of an optimal assignment. ∎

The greedy algorithm generalizes the greedy algorithm in the bounded delay model, presented by Kesselman et al. [69]. They show that there exists an arrival sequence for which greedy attains a competitive ratio of 2, and therefore our analysis above is tight.

A special case is when the time loss functions are linear with the delay, *i.e.*, $\mathrm{val}(p,t) = \mathrm{val}(p) - c_p \cdot \mathrm{delay}(p)$, where c_p is p's loss per unit of time.

We next show that in the case of heterogenous linear loss function, the competitive ratio can be improved to $e/(e-1)$ using randomization. We present algorithm "Randomized Greatest Loss First"(Rglf) — every time the algorithm sends a packet it selects from the packet which will suffer the highest loss, from the set of packets with a value that exceeds a certain threshold, chosen randomly. Algorithm Rglf and its analysis are inspired by the Rmix algorithm presented and analyzed by Chin et al.[25].

Algorithm Rglf: At every time t algorithm Rglf selects randomly and uniformly a value x in the interval $[-1,0]$. Let X_t denote the content of Rglf buffer at time t, *i.e.*, every packet that has strictly positive value at time t, and that was not sent previously. Let h denote the packet in X_t with the current highest value $\mathrm{val}(h,t) = \max_{j \in X_t}\{\mathrm{val}(j,t)\}$. The algorithm selects the packet with the greatest loss factor, from the set of packets with value at least $e^x w_{h,t}$. If several packets have a value that exceeds the threshold, and their loss factor is identical to the highest loss factor, then Rglf selects the one with the current lowest value.

The analysis of Rglf uses a potential function, that maps buffer configurations[2] to the real numbers. We use the following lemma from [25], regarding the analysis of online scheduling algorithm.

Lemma 6.16. *[from [25]] Let $\mathcal{A}$ be an online algorithm for scheduling online packets and let ψ be a potential function such that a configuration with no pending packets is mapped to 0. If ψ satisfies at each step*

$$\beta \Delta \mathcal{A} \geq \Delta OPT + \Delta \psi, \tag{6.7}$$

[2] A configuration at event E consists of the content of the online algorithm buffer, and the content of the optimal schedule buffer after the last event that precedes E and before E.

where $\Delta\psi$ represents the change in the potential, and $\Delta\mathcal{A}, \Delta\mathrm{OPT}$ represent the $\mathcal{A}$'s, and OPT's change in gain in this step. Then $\mathcal{A}$ is β-competitive.

To show this lemma we sum the $\Delta\mathrm{OPT}, \Delta\mathcal{A}$, over all events. Notice that because of linearity of expectation, the same argument holds for randomized algorithms for which the expected change in the algorithm $\mathrm{E}[\Delta\psi]$, satisfies condition 6.7.

Theorem 6.17. *The competitive ratio of algorithm* RGLF *is at most $e/(e-1)$.*

Proof. The proof follows the line of proof of the competitive ratio of algorithms RMix, introduced in [25] by Chin et al. .

At a given time step t, let Y, X denote the buffers of optimal schedule and the online algorithm respectively. We define the potential function

$$\psi = \sum_{j \in Y \setminus X} \mathrm{val}(j, t).$$

By Lemma 6.16, it suffices to show that at a time step t at which OPT schedules a packet j, $\mathrm{E}[\frac{e}{e-1}\mathrm{val}(f,t) - \Delta\psi] > \mathrm{val}(j,t)$, where f is a random variable denoting the packet selected by RGLF.

For the entire proof we fix a time slot t. Arrival event do not change the potential function, since every arriving packet immediately joints X. Expiration of packets also does not affect the potential, since no expired packet every belongs to Y.

In case that $j \in Y \setminus X$. The potential increases by at most the value of f at the next time slot that is $\mathrm{val}(f, t+1) = \mathrm{val}(f,t) - c_f$, and decreases by $\mathrm{val}(j,t)$. Thus, $\Delta\psi \leq \mathrm{val}(f,t) - \mathrm{val}(j,t)$, and therefore

$$\mathrm{E}[\frac{e}{e-1}\mathrm{val}(f,t) - \Delta\psi] \geq \mathrm{E}[\frac{1}{e-1}\mathrm{val}(f,t) + \mathrm{val}(j,t)] \geq \mathrm{val}(j,t),$$

and we are done

Otherwise, suppose that $j \in Y \cap X$. In this case we can immediately tell that $\Delta\psi \leq \mathrm{val}(f,t) - c(f) < \mathrm{val} f, t$, as packet $j \in X$ is not part of the sum in the potential.

In case that $\mathrm{val}(j,t) \geq e^x \mathrm{val}(h,t)$, then j would have been sent by RGLF, if $c(j) \geq c(i)$ for every $i \in X$. We can assume without loss of generality, that if $c_j = c_f$ then OPT schedules f at time t, *i.e.*, $f = j$, otherwise, we could modify OPT to replace the send order of f and j. This is possible since f was selected to be the packet with the lowest current value over all packets i with $c_f = c_i$. Therefore, changing the order is possible, because if f is scheduled by OPT for later then t, then so could be j. Therefore, either

(i) $f = j$. In this case f will not belong to Y in the next step, or,

(ii) $f \neq j$. In which case we can assume without loss of generality that $c_f > c_j$, which again means that f does not belong to Y, as it will not be sent by OPT (otherwise, OPT could gain by switching the order between f and j).

Thus, $\Delta\psi = 0$. As a result,

$$\begin{aligned}
\mathrm{E}[\frac{e}{e-1}\mathrm{val}(f,t) - \Delta\psi] &= \frac{1}{e-1}E[\mathrm{val}(f,t)] + E[\mathrm{val}(f,t) - \Delta\psi] \\
&\geq \int_{-1}^{0} e^x \mathrm{val}(h,t)dx + \int_{-1}^{\log\frac{\mathrm{val}(j,t)}{\mathrm{val}(h,t)}} (\mathrm{val}(f,t) - 0)\, dx \\
&+ \int_{\log\frac{\mathrm{val}(j,t)}{\mathrm{val}(h,t)}}^{0} (\mathrm{val}(f,t) - \Delta\psi)\, dx \\
&\geq \frac{1}{e-1}\left(1 - \frac{1}{e}\right)\mathrm{val}(h,t) + \int_{-1}^{\log\frac{\mathrm{val}(j,t)}{\mathrm{val}(h,t)}} e^x \mathrm{val}(h,t)dx \\
&= \frac{1}{e}\mathrm{val}(h,t) + \left(\frac{\mathrm{val}(j,t)}{\mathrm{val}(h,t)} - e^{-1}\right)\mathrm{val}(h,t) \\
&= \mathrm{val}(j,t)
\end{aligned}$$

This completes the proof. ∎

Last we show that unlike the case of homogenous time loss, when time-loss is heterogenous, the competitive ratio of every algorithm is bounded away from 1, even if preemption is allowed and the FIFO order need not be kept.

Claim 6.18. *The competitive ratio of every deterministic online algorithm is at least $\sqrt{6}/2 \approx 1.224$. The competitive ratio of every algorithm (deterministic or randomized) is at least 1.101.*

Proof. We construct a sequence σ for a deterministic online algorithm ON. The sequence σ begins with two arrivals at the first time slot of packet p with $\mathrm{val}(p) = 2$ and $c(p) = 1$, and packet q with $\mathrm{val}(q) = c(q) = \sqrt{6}-1$. If at the first send event ON sends p then the sequence ends. In this case the competitive ratio is $\mathrm{OPT}(\sigma)/\mathrm{ON}(\sigma) = \sqrt{6}/2$. If ON sends q, then σ has an additional arrival of packet r at the second time slot, with $\mathrm{val}(r) = c(r) = 1$. In this case $\mathrm{OPT}(\sigma)\mathrm{ON}(\sigma) = 3/\sqrt{6} = \sqrt{6}/2$. Thus, the competitive ratio of every deterministic online algorithm ON is at least $\sqrt{6}/2$.

A lower bound of 1.10 on every (possibly randomized) algorithm, can be derived from the same construction combined with Yao's principle[18]. ∎

6.5 Regulating Selfish Packets

We now point to a connection between our work and Naor's model [81], and state our results in terms of social welfare given selfish customers. If incoming packets are regarded as selfish decision makers, then the simple threshold policy is a dominating strategy. Customers join the queue if and only if its length is less than their value. In the worst case, the stream is such that the queue size is always slightly less than the value of the incoming packet, and hence, the profit for a customer is always 0. From the perspective of [81], Theorem 6.5 can be interpreted as a proposition to levy a toll of $1/\phi \cdot R$ at the entry point of a queue. This ensures a $1/\phi$ approximation of the optimal social welfare. Theorem 6.5 tells us no other online payment scheme can achieve a better guarantee in the worst case.

In the case of heterogenous customers, the value of a packet can be regarded as its type, and the problem of levying tolls can be seen as a mechanism design issue. Theorem 6.8 can be interpreted as a payment mechanism, where the price for a customer to enter the queue equals the queue size on his arrival. Clearly, this mechanism is truthful. Moreover, customers make no bid, only decide on whether to join the queue or to balk. It follows from the competitive ratio of 8 for DT that this mechanism guarantees an approximation ratio of 8 on the optimal social welfare, in the worst case.

Revenue Maximization. In this chapter we consider a "social welfare" viewpoint, as our goal is to maximize the aggregated gain of individuals. To enforce social welfare joining policies we suggested that tolls should be levied on customers accepting service. When the queue manager charges a fee for the service given, a natural question is what policy/ pricing scheme would maximize the manager's revenue? Naor [81] consider this question in his stochastic model with homogeneous customers, and shows that the fee charged by a profit maximizing queue manager is higher than the optimal social welfare maximizing fee.

In the corresponding online setting (homogeneous packets; FIFO queue) we could charge an admission fee of R/ϕ, to impose our threshold strategy. Alternatively, the queue manager could charges at time t the minimum between $R - B(t)$, and R/ϕ. This would extract the entire value from accepted packets, and would guarantee a competitive ratio of ϕ on the revenue of the manager, which following Theorem 6.5, is the best possible.

This pricing scheme is based on the fact that the owner knows that true value R of every packet. The question of designing an optimal revenue maximizing pricing scheme becomes much more challengeable in the heterogeneous value model, and remains open.

6.6 An Optimal Offline Algorithm

In this section we describe efficient offline algorithms for maximizing the sum of packet benefits (alternately, in the context of utilities and selfish agents, "maximizing the social welfare given the true agent utilities"). We describe algorithms for heterogenous packets. Ergo, they are trivially applicable in case of homogenous packets as well. We observe that the offline problem can be solved via bipartite matching in $O(n^3)$ time complexity, where n is the number of arriving packets in the event sequence. We also show a greedy algorithm that also gives the optimal sum of benefits, and requires only $O(n \log n)$ time complexity.

A natural approach to solving the offline problem of maximizing benefit from heterogenous packets is to use weighted bipartite matching. One could introduce a vertex for every packet and every time slot. Add edges from packets p to time slots $t > \text{arrive}(p)$, of weight $\text{val}(p) - \lfloor t - \text{arrive}(p) \rfloor$. Now, compute a maximal weighted matching. Packets that are matched to time slots will be accepted by the queue policy, packets that are not matched will be rejected. We could interpret an edge matching packet p to time slot t as though this means that packet p is to be transmitted at time t and this implies some arbitrary non-preemptive queue regime. Fortunately, it follows from Observation 6.20 that using the FIFO queue regime will result in the same benefit[3].

Given an event sequence, a transmission schedule is a mapping, m, from the arriving packets of the event sequence to the integers ≥ 1 or the special symbol $\emptyset$, such that

1. If packet p is mapped to $m(p) \neq \emptyset$ then $\text{arrive}(p) < m(p)$, this can be interpreted as saying that packet p is to be transmitted at time slot $m(p)$. We interpret $m(p) = \emptyset$ as indicating that packet p is rejected.

2. No two packets are mapped to the same integer.

[3] We remark that this bipartite matching approach can be used for much more general latency sensitivity scenarios, but then Observation 6.20 may become inapplicable.

Given an event sequence σ, we define the total benefit of the transmission schedule m to be $\sum_{p\in S} \text{benefit}(p)$, where S is the set of all packets p in σ such that $m(p) \neq \emptyset$. Recall that if $m(p) \neq \emptyset$ then $\text{benefit}(p) = \text{val}(p) - \lfloor m(p) - \text{arrive}(p) \rfloor$. Given an event sequence σ, a queue policy π, and a queue regime $\mathcal{R}$, the pair $(\pi, \mathcal{R})$ jointly determines a unique transmission schedule $m : \sigma \mapsto Z^+ \cup \{\emptyset\}$[4].

Observation 6.19. *Given an event sequence σ, and a transmission schedule m, choose any two packets, p and q, such that $m(p) \neq \emptyset$ and $m(q) \neq \emptyset$, and such that the arrival times of both p and q are prior to the earlier transmission (i.e.,* $\max(\text{arrive}(p), \text{arrive}(q)) < \min(m(p), m(q))$*). Now, define a new transmission schedule $m' = m$, except in that $m'(p) = m(q)$ and $m'(q) = m(p)$. (I.e., m' switches the transmission times of packets p and q). Then, the total benefit from transmission schedules m and m' is the same.*

Given an event sequence σ and a transmission schedule m, let m' be another transmission schedule reachable from m by successive applications of transmission time swaps as described in Observation 6.19. Let π be a queue policy that determines m (in conjunction with some queue regime $\mathcal{R}$). Note that if some packet p has negative benefit in schedule m' (*i.e.*, $\text{benefit}(p) = \text{val}(p) - \lfloor m'(p) - \text{arrive}(p) \rfloor < 0$), then the queue policy π is clearly non optimal, since π would have benefited by rejecting packet p.

To derive an optimal offline policy for non preemptive FIFO queues, we start by allowing preemptive and arbitrary queue regimes. The following observation states that while preemption may add considerable strength to online algorithms, it does not impact the optimal (offline) solution. It also states that the queue regime is irrelevant in the context of non-preemptive queue regimes.

Observation 6.20.

1. *The maximal total benefit is the same irrespective of whether the queue regime allows preemption or not.*
2. *For non-preemptive queue regimes, the total benefit from an event sequence depends only on the admission policy and not on queue regime.*

We consider the following algorithm, GREEDY, which defines both the queue policy and regime: initially, GREEDY accepts every incoming packet. On a send event, GREEDY transmits the packet that currently possesses the highest benefit (ties are broken arbitrarily), and preempts packets with negative or zero benefit.

[4] Likewise, given an event sequence σ and a transmission schedule $m : \sigma \mapsto Z^+ \cup \{\emptyset\}$, there exist (possibly many) pairs of (queue policy,queue regime) that determine m.

Theorem 6.21. *For any event sequence σ, $\text{GREEDY}(\sigma) = \text{OPT}(\sigma)$, GREEDY has $O(n \log n)$ time complexity, where n is the number of arriving packets in σ.*

GREEDY defines both a queue policy and a queue regime, the queue regime used by GREEDY is not FIFO. However, it follows from Observation 6.20 that we could define a queue policy , OPT, that accepts the same packets eventually transmitted by GREEDY, along with any non-preemptive queue regime, and achieve maximal benefit.

6.7 Future Research Directions

We outline a few directions for a future research:

(i) We conjecture that the competitive ratio of deterministic algorithms for heterogeneous packets is $\phi^3 \approx 4.23$.

(ii) Although randomness did not help in the homogenous case, it seems possible that randomized algorithms can beat the deterministic lower bound of ϕ^3 for heterogeneous packets.

(iii) A major goal would be to control a queue with intrinsic packet values and heterogenous value loss functions.

Bibliography

[1] J. Abernethy, E. Hazan, and A. Rakhlin. Competing in the dark: An efficient algorithm for bandit linear optimization. In *COLT*, pages 263–274, 2008.

[2] N. Abramson. The aloha system - another alternative for computer communications. *in Proc. of the Fall Joint Computer Conference*, 14(4):281–285, 1970.

[3] I. Adler and P. Naor. Social optimization versus self-optimization in waiting lines. Technical report, 1969.

[4] E. Altman, R. El Azouzi, and T. Jiménez. Slotted aloha as a game with partial information. *Computer Networks*, 45(6):701–713, 2004.

[5] E. Altman, D. Barman, R. El Azouzi, and T. Jiménez. A game theoretic approach for delay minimization in slotted aloha.

[6] E. Altman, T. Basar, T. Jimnez, and N. Shimkin. Competitive routing in networks with polynomial cost. *IEEE Transactions on Automatic Control*, 47:92–96, 2000.

[7] N. Andelman, Y. Mansour, and A. Zhu. A competitive queueing policies for QoS switches. In *Proc. of 14th ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 761–770, 2003.

[8] E. Anshelevich, A. Dasgupta, J. M. Kleinberg, É. Tardos, T. Wexler, and T. Roughgarden. The price of stability for network design with fair cost allocation. In *FOCS*, pages 295–304, 2004.

[9] A. Archer and E. Tardos. Truthful mechanisms for one-parameter agents. In *FOCS*, pages 482–491, 2001.

[10] Sanjeev Arora and Bo Brinkman. A randomized online algorithm for bandwidth utilization. In *SODA '02: Proceedings of the thirteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 535–539, 2002.

[11] P. Auer, N. Cesa-Bianchi, Y. Freund, and R. E. Schapire. The nonstochastic multi-armed bandit problem. *SIAM Journal on Computing*, 32(1):48–77, 2002.

[12] B. Awerbuch, Y. Azar, and A. Epstein. Large the price of routing unsplittable flow. In *STOC '05: Proceedings of the thirty-seventh annual ACM symposium on Theory of computing*, pages 57–66. ACM, 2005.

[13] B. Awerbuch and R. Kleinberg. Adaptive Routing With End-to-End Feedback: Distributed Learning and Geometric Approaches. In *Proceedings of the 36th ACM Symposium on Theory of Computing*, pages 45 – 53, 2004.

[14] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss. An architecture for differentiated service, 1998.

[15] A. Blum, E. Even-Dar, and K. Ligett. Routing without regret: On convergence to nash equilibria of regret-minimizing algorithms in routing games. In *PODC*, pages 45–52, 2006.

[16] A. Blum, K. Ligett M. Hajiaghay and, and A. Roth. Regret minimization and the price of total anarchy. In *STOC '08: STOC '05: Proceedings of the Fortieth Annual ACM symposium on Theory of computing*, pages 373–382, 2008.

[17] A. Blum and Y. Mansour. From external to internal regret. *Journal of Machine Learning Research (JMLR)*, 8:1307–1324, 2007.

[18] A. Borodin and R. El-Yaniv. *Online Computation and Competitive Analysis.* Cambridge University Press, 1998.

[19] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

[20] G.W. Brown. Iterative solution of games by fictitious play. *Activity analysis of production and allocation*, 13:374–376, August 1951.

[21] N. Buchbinder and J. Naor. Online primal-dual algorithms for covering and packing problems. In *ESA*, pages 689–701, 2005.

[22] I. Capetanakis. Tree algorithms for packet broadcast channels. *IEEE Trans. on Information Theory*, IT-25(5):505–515, 1979.

[23] S. Catoni and S. Pallottino. Traffic equilibrium paradoxes. *Transportation Sci.*, 25 (3):240–244, (1991).

[24] N. Cesa-Bianchi and G. Lugosi. Potential-based algorithms in on-line prediction and game theory. *Machine Learning*, 51(3):239–261, 2003.

[25] F. Y. L. Chin, M. Chrobak, S. P. Y. Fung, W. Jawor, J. Sgall, and T. Tichy. Online competitive algorithms for maximizing weighted throughput of unit jobs. *Journal of Discrete Algorithms*, 4(2):255–276, 2006.

[26] F. Y. L. Chin and S. P. Y. Fung. Online scheduling with partial job values: Does timesharing or randomization help? *Algorithmica*, 37(3):149–164, 2003.

[27] D. Clark and J. Wroclawski. An approach to service allocation in the internet, July 1997.

[28] R. Cole, S. Dobzinski, and L. Fleischer. Prompt mechanisms for online auctions. In *SAGT*, pages 170–181, 2008.

[29] A. A. Cournot. Recherches sur les principes mathmatiques de la thorie des richesses.(english translation: Researches into the mathematical principles of the theory of wealth). 1838.

[30] P. Cramton, Y. Shoham, and R. Steinberg. *Combinatorial Auctions*. The MIT Press, 2006.

[31] A. Czumaj and B. Vöcking. Tight bounds for worst-case equilibria. In *SODA '02: Proceedings of the thirteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 413–420, Philadelphia, PA, USA, 2002. Society for Industrial and Applied Mathematics.

[32] V. Dani and T. P. Hayes. Robbing the Bandit: Less Regret in Online Geometric Optimization Against an Adaptive Adversary. In *Proceedings of the 17th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 937 – 943, 2006.

[33] B. Edelman, M. Ostrovsky, and M. Schwarz. Internet advertising and the generalized second-price auction: Selling billions of dollars worth of keywords. *American Economic Review*, 97(1):242–259, March 2007.

[34] M. Englert and M. Westermann. Considering sup-pressed packets improves buffer management in qos switches. In *Proc. of 18th ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2007.

[35] E. Even-Dar, Y. Mansour, and U. Nadav. On the convergence of regret minimization dynamics in concave games. In *STOC '09: Proceedings of the forty-first annual ACM symposium on Theory of computing*, pages 523–532, 2009.

[36] A. Fabrikant, A. Luthra, E. Maneva, C. H. Papadimitriou, and S. Shenker. On a network creation game. In *Proceedings of the twenty-second annual symposium on Principles of distributed computing*, 2003.

[37] J. Feigenbaum, C. H. Papadimitriou, R. Sami, and S. Shenker. A BGP-based mechanism for lowest-cost routing. In *PODC*, pages 173–182, 2002.

[38] Moran Feldman. Private communication, 2008.

[39] A. Fiat, Y. Mansour, and U. Nadav. Efficient contention resolution protocols for selfish agents. In *SODA '07: Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 179–188, 2007.

[40] A. Fiat, Y. Mansour, and U. Nadav. Competitive queue management for latency sensitive packets. In *SODA '08: Proceedings of the nineteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 228–237, 2008.

[41] A. Flaxman, A. T. Kalai, and B. McMahan. Online convex optimization in the bandit setting: Gradient descent without a gradient. In *Proceedings of the Sixteenth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 373–382, 2005.

[42] S. Floyd and Jacobson V. Random early detection gateways for congestion avoidance. *IEEE/ACM Transactions on Networking*, 1:397–413, 1993.

[43] D. Foster and S. M. Kakade. Deterministic calibration and nash equilibrium. In *COLT*, pages 33–48, 2004.

[44] D. Foster and R. Vohra. Regret in the on-line decision problem. *Games and Economic Behavior*, 21:40–55, 1997.

[45] D. Foster and H. P. Young. Learning, hypothesis testing, and Nash equilibrium. *Games and Economic Behavior*, 45:73–96, 2003.

[46] D. Foster and H. P. Young. Regret testing: Learning to play Nash equilibrium without knowing you have an opponent. *Theoretical Economics*, 1:341–367, 2006.

[47] Y. Freund and R. Schapire. Adaptive game playing using multiplicative weights. *Games and Economic Behavior*, 29:79–103, 1999.

[48] Yoav Freund and Robert E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. In *Euro-COLT*, pages 23–37. Springer-Verlag, 1995.

[49] D. Fudenberg and D. K. Levine. *The theory of learning in games.* MIT press, 1998.

[50] F. Germano and G. Lugosi. Global Nash convergence of Foster and Young's regret testing. *Games and Economic Behavior*, 60:135–154, 2007.

[51] B. Hajek and G. Gopalakrishnan. Do greedy autonomous systems make for a sensible internet?, 2002. presented at the Conference on Stochastic Networks, Stanford University.

[52] J. Hannan. Approximation to bayes risk in repeated plays. In M. Dresher, A. Tucker, and P. Wolfe, editors, *Contributions to the Theory of Games*, volume 3, pages 97–139. Princeton University Press, 1957.

[53] G. Hardin. The tragedy of the commons. *Science*, 162(3859):1243–1248, 1968.

[54] P.T. Harker. Multiple equilibrium behaviors on networks. *Transportation Sci.*, 22(1):39–46, (1988.

[55] S. Hart and Y. Mansour. The communication complexity of uncoupled Nash equilibrium procedures. In *STOC*, pages 345–353, 2007.

[56] S. Hart and A. Mas-Colell. A simple adaptive procedure leading to correlated equilibrium. *Econometrica*, 68:1127–1150, 2000.

[57] S. Hart and A. Mas-Colell. Stochastic uncoupled dynamics and Nash equilibrium. *Games and Economice Behavior*, 57(2):286–303, 2006.

[58] R. Hassin. On the optimality of first come last served queues. *Econometrica*, 53(1):201–202, 1985.

[59] R. Hassin and M. Haviv. *To Queue or not to Queue: Equilibrium Behavior in Queueing Systems*. Kluwer Apademic Publishers, 2003.

[60] E. Hazan, A. Agarwal, and S. Kale. Logarithmic regret algorithms for online convex optimization. *Machine Learning*, 69(2-3):169–192, 2007.

[61] E. Hazan, E. Kalai, S. Kale, and A. Agarwal. Logarithmic regret algorithms for online convex optimization. In *COLT*, pages 499–513, 2006.

[62] V. Jacobson. Congestion avoidance and control. *Proceedings of SIGCOMM*, 13:1176–1188, August 1988.

[63] R. Joahri and J. Tsitsiklis. Efficiency loss in resource allocation games. *Mathematics of Operations Research*, 29(3):407–435, 2004.

[64] A. Kalai and S. Vempala. Efficient algorithms for on-line optimization. *Journal of Computer Systems and Science (JCSS)*, 71(3):291–307, 2005. (A preliminary version appeared in COLT 2003.).

[65] R. M. Karp, E. Koutsoupias, C. H. Papadimitriou, and S. Shenker. Optimization problems in congestion control. In *FOCS '00: 41st Annual Symposium on Foundations of Computer Science*, pages 66–74, 2000.

[66] F. Kelly. Charging and rate control for elastic traffic. *European Transactions on Telecommunications*, 8:33–37, 1997.

[67] F. Kelly, A. Maulloo, and D. Tan. Rate control in communication networks: shadow prices, proportional fairness and stability. In *Journal of the Operational Research Society*, volume 49, 1998.

[68] S. Keshav. *Congestion control in computer networks*. PhD thesis, Berkeley, CA, USA, 1992.

[69] A. Kesselman, Z. Lotker, Y. Mansour, B. Patt-Shamir, B. Schieber, and M. Sviridenko. Buffer overflow management in QoS switches. *SIAM J. on Computing*, 33(3):563–583, 2004.

[70] Robert Kleinberg, Georgios Piliouras, and Éva Tardos. Multiplicative updates outperform generic no-regret learning in congestion games: extended abstract. In *STOC '09: Proceedings of the forty-first annual ACM symposium on Theory of computing*, pages 533–542, 2009.

[71] N. C. Knudsen. Individual and social optimization in a multiserver queue with a general cost-benefit structure. *Econometrica*, 40(3):515–28, May 1972.

[72] E. Koutsoupias and C. H. Papadimitriou. Worst-case equilibria. In *Proceedings of the 16th Annual Symposium on Theoretical Aspects of Computer Science*, pages 404–413, 1999.

[73] F. Li, J. Sethuraman, and C. Stein. An optimal online algorithm for packet scheduling with agreeable deadlines. In *Proc. of 16th ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 801–802, 2005.

[74] F. Li, J. Sethuraman, and C. Stein. Better online buffer management. In *Proceedings of the Eighteenth ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 199–208, 2007.

[75] S. H. Low and D. E. Lapsley. Optimization flow control, i: Basic algorithm and convergence. *IEEE/ACM Transactions on Networking*, 7(6):861–874, 1999.

[76] A. MacKenzie and S. B. Wicker. Stability of multipacket slotted aloha with selfish users and perfect information. In *INFOCOM*, 2003.

[77] A. Mas-Colell, J. Green, and M. D. Whinston. *Microeconomic Theory*. Oxford University Press, 1995.

[78] I. Menache and N. Shimkin. Efficient rate-constrained nash equilibrium in collision channels with state information. In *INFOCOM*, pages 403–411. IEEE, 2008.

[79] J. Mosely and P.A. Humblet. A class of efficient contention resolution algorithms for multiple access channels. *IEEE Trans. on Communications*, COM-33(2):145–151, 1985.

[80] J. Naor, D. Raz, and G. Scalosub. Homogeneous interference game in wireless networks. In *WINE '08: Proceedings of the 4th International Workshop on Internet and Network Economics*, pages 514–521, Berlin, Heidelberg, 2008. Springer-Verlag.

[81] P. Naor. The regulation of queue size by levying tolls. *Econometrica*, 37(1):15–24, 1969.

[82] J. F. Nash. Equilibrium points in n-person games. In *Proceedings of the National Academy of Sciences of the United States of America*, 1950.

[83] N. Nisan and A. Ronen. Algorithmic mechanism design (extended abstract). In *STOC '99: Proceedings of the thirty-first annual ACM symposium on Theory of computing*, pages 129–140, 1999.

[84] N. Nisan, T. Roughgarden, E. Tardos, and V.V. Vazirani, editors. *Algorithmic Game Theory*. Cambridge University Press, 2007.

[85] M. Osborne and A. Rubinstein. *A Course in Game Theory*. Mit., New York, NY, USA, 1994.

[86] D. C. Parkes. Online mechanisms. In Noam Nisan, Tim Roughgarden, Eva Tardos, and Vijay Vazirani, editors, *Algorithmic Game Theory*, chapter 16. Cambridge University Press, 2007.

[87] M. L. Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley & Sons, Inc., New York, NY, USA, 1994.

[88] R. Rom and M. Sidi. *Multiple access protocols: performance and analysis*. Springer-Verlag New York, Inc., New York, NY, USA, 1990.

[89] J. B. Rosen. Existence and uniqueness of equilibrium points for concave n-person games. *Econometrica*, 33(3):520–534, 1965.

[90] R. W. Rosenthal. A class of games possessing pure-strategy nash equilibria. *Journal of Game Theory*, 2:65–67, 1973.

[91] T. Roughgarden. *Selfish Routing and the Price of Anarchy*. MIT press, 2005.

[92] T. Roughgarden. Selfish routing with atomic players. In *Proceedings of the Sixteenth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1184 – 1185, 2005.

[93] T. Roughgarden and E. Tardos. How bad is selfish routing? *Journal of the ACM*, 49(2):236 – 259, 2002.

[94] T. Roughgarden and E. Tardos. Introduction to the inefficiency of equilibria. In N. Nisan, T. Roughgarden, E. Tardos, and V.V. Vazirani, editors, *Algorithmic Game Theory*. Cambridge University Press, 2007.

[95] S. Shenker. Fundamental design issues for the future internet. *IEEE Journal on Selected Areas in Communication*, 13(7), September 1995.

[96] R. D. Theocharis. On the stability of the cournot solution on the oligopoly problem. *The Review of Economic Studies.*, 27(2):133–134, 1960.

[97] B.S. Tsybakov and N. B. Likhanov. Upper bound on the capacity of a random multiple-access system. *Problemy Peredachi Informatsii*, 23(3):64–78, 1987.

[98] B.S. Tsybakov and V.A. Mikhailov. Free synchronous packet access in a broadcast channel with feedback. *Probl. Information Transmission*, 14(4):259–280, (October-December 1978.

[99] A. Veres and M. Boda. The chaotic nature of tcp congestion control. *INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, 3, 2000.

[100] N. D. Vvedenskaya and M.S. Pinsker. Non-optimality of the part-and-try algorithm. *In Abstracts of the International Workshop on Convolutional Codes, Multiuser Communication, Sochi, USSR*, pages 141–144, 1983.

[101] U. Yechiali. On optimal balking rules and toll charges in the gi/m/1 queuing process. *OPERATIONS RESEARCH*, 19(2):349–370, March-April 1971.

[102] H. P. Young. *Strategic Learning and Its Limits*. Oxford University Press, 2004.

[103] M. Zinkevich. Online convex programming and generalized infitesimal gradient ascent. In *ICML*, pages 928–936, 2003.

[104] M. Zinkevich. Theoretical guarantees for algorithms in multi-agent settings. Technical Report CMU-CS-04-161,, Carnegie Mellon University, 2004.