Learning and Implementation on the Internet

Eric J. Friedman*
Department of Economics, Rutgers University
New Brunswick, NJ 08903

Scott Shenker
Xerox PARC
3333 Coyote Hill Road, Palo Alto, CA 94304-1314

August 14, 1998

Abstract

We address the problem of learning and implementation on the Internet. When agents play repeated games in distributed environments like the Internet, they have very limited *a priori* information about the other players and the payoff matrix, and the play can be highly asynchronous. Consequently, standard solution concepts like Nash equilibria, or even the serially undominated set, do not apply in such a setting. To construct more appropriate solution concepts, we first describe the essential properties that constitute "reasonable" learning behavior in distributed environments. We then study the convergence behavior of such algorithms; these results lead us to propose rather non traditional solutions concepts for this context. Finally, we discuss implementation of social choice functions with these solution concepts.

*We would like to thank Roger Klein and Hervé Moulin for useful discussions and seminar participants at Princeton, Rutgers, Stonybrook and UBC, for helpful comments. This research was supported by NSF Grant ANI-9730162. Email: friedman@econ.rutgers.edu, shenker@parc.xerox.com

1 Introduction

The Internet is rapidly becoming a centerpiece of the global telecommunications infrastructure, and someday it may well provide all of our telecommunication needs. In this paper we consider the Internet as an exercise in resource sharing, where the sharing occurs on several different levels. Most importantly, Internet users share access to the underlying transmission facilities themselves. With the best-effort nature of the Internet, where resources are not reserved and all packets are serviced on a first-come-first-serve basis, one user's usage can affect the quality of service seen by another user.[1] In addition, the Internet provides a seamless way of accessing remote services, such as databases or web servers, which are themselves examples of shared resources where usage can induce congestion. For example, delays on the World-Wide-Web have increased significantly in recent years – it is now sometimes waggishly referred to as the "World-Wide-Wait" – and service providers such as America Online have faced lawsuits over their access delays. Both of these are cases where overuse has resulted in deteriorating service quality for all users.

In each case, aggressive applications (or users) get more than an equal share of these shared facilities, and so the Internet is likely to be a place where noncooperative game theory is particularly relevant. For instance, web browsers that open more TCP[2] connections receive more bandwidth (at the expense of less opportunistic users of the Internet).[3] Similarly, users that modify their TCP implementation to be less responsive when congestion is detected can obtain much larger shares of the bandwidth (Demers et al. 1990).

[1]This effect does not occur on telephone networks because the underlying transmission facilities are not shared on a packet-by-packet basis; bandwidth is reserved for each call and so the quality of service perceived by a particular user is independent of the presence of other callers.

[2]TCP stands for Transmission Control Protocol, and this is the protocol that governs the bandwidth usage in data transfers. In particular, TCP is designed so that flows slow down their rate of transmission when they detect congestion.

[3]In the Netscape Navigator browser, the maximal number of TCP connections can be set by the user, so that this form of "greediness" is under user control!

For the Internet's architecture to be viable in the long-term, it must not be vulnerable to such greedy users, and thus it must be designed with incentives in mind. Network architects are increasingly addressing the incentive properties of their designs. For example, McCanne et al. (1996) discuss the incentive issues in packet dropping algorithms and its implications for layered multicast (see Bajaj et al. 1998 for a continuation of this line of investigation); Nagle (1985) was the first to explore the incentive issues inherent in packet scheduling in network routers, and this has been the focus of much subsequent research (see, for example, Sanders 1988, Demers et al. 1990, Shenker 1990 and 1995, Korilis and Lazar 1993, Korilis et al. 1995); Resnick et al. (1996) have proposed market-based solutions to the problem of network address allocation and route advertisements. Networks with multiple qualities of service raise interesting incentive issues, and this has promoted much of the recent interest in pricing and accounting for computer networks (a few examples include Cocchi et al. 1993, Clark et al. 1992, MacKie-Mason and Varian 1993 and 1994, Murphy and Murphy 1994, Mendelson and Whang 1990).

For similar reasons, many theorists have begun applying game theory to the Internet (see, for example, Ferguson 1989, Ferguson et al. 1989, Gupta et al. 1994, Hsiao and Lazar 1988, Korilis et al. 1995 and 1996). Most of these analyses assume that the appropriate solution concept – the set of asymptotic plays in a repeated game – is contained within the set of Nash equilibria. To the contrary, in this paper we argue that Nash equilibria are not necessarily achieved as a result of learning in the Internet setting and that, in fact, distributed settings like the Internet require a dramatically different solution concept.

Because of the Internet's increasing role in the telecommunications infrastructure, it is important that we achieve socially desirable allocations of service in the Internet. This will require understanding the nature of learning and convergence in the Internet, and other distributed settings, so that we can identify the appropriate solution concept. Learning and convergence, and its implications for mechanism design in the Internet, is the subject of this paper.

For a concrete example of the incentive issues with which we are concerned, consider the scenario (which is more fully described in Shenker 1995) where several Internet users are simultaneously sending data across a particular link. The delay experienced by the packets is a function of the load – the bandwidth consumed by various users – on the link. Each user's utility function U_i depends on her average bandwidth (transmission rate), r_i, and on the average queuing or congestion experienced by her packets, c_i. Users control their bandwidth usage r_i, and the network determines the vector of average queuing c as a function of the set of bandwidths r: i.e., $c = C(r)$ where the function $C(\cdot)$ reflects the particular packet scheduling algorithm used by the network (and must obey the sum rule that $\sum_i C_i(r) = f(\sum_i r_i)$ for some constraint function f because the overall average queue length is independent of the order in which packets are served). This "congestion game", where each player's usage can impose delay on other players, can be modeled as a normal form game, with the bandwidths r_i being the actions and the payoffs given by $U_i(r_i, C_i(r))$. The equilibria or, more generally, the solution concept of this congestion game will determine the allocation of network bandwidth among these users. Since network designers can choose the scheduling algorithm $C(\cdot)$ in order to attain some socially desirable outcome, the solution concept of this congestion game has significant practical ramifications.

This congestion game also arises in many other settings. For instance, r_i could be the usage level of a shared database (such as a video or text library) or web server with c_i being be the processing delay, or r_i could be the average time connected to an online service with c_i being the expected time required to connect. (See, Friedman, 1997 for a discussion of this and other games arising on the Internet.) These examples suggest that there are many game-like situations arising in *distributed* systems like the Internet. We call them distributed systems because the users are geographically dispersed and are accessing the resource through the network. The 'games' in these distributed systems share the feature that the agents interact only through their joint use of a shared resource; for instance, the only form of interaction between users in the congestion game is that their packets happen to collide somewhere inside the network. Thus, it is quite likely that the agents have little or no information about each other. Moreover, the users probably know very little about the detailed nature (e.g., capacity, latency, etc.) of the resource itself; to use the congestion game again as a specific example, users have little knowledge of the underlying network topology and characteristics so they can't always distinguish between delays due to the characteristics of the underlying network (e.g., speed-of-light delays in transmission links) and delays due to the behavior of other users (e.g., queuing delays in routers).

In this paper we ask two questions: 1) What is the appropriate solution concept for the congestion game and other games that arise in distributed settings? 2) Given this solution concept, can we design scheduling or sharing algorithms to achieve the allocations we desire? If the congestion game were a canonical one-shot game with common knowledge, then one could invoke standard solution concepts such as Nash equilibria or the rationalizable set. However, the congestion game is neither a one-shot game nor one with common knowledge. Many data transmissions persist for a significant period of time, and the users are able to adjust their bandwidth at any point while transmitting. Thus, the congestion game should be modeled as a repeated game rather than a one-shot game. Moreover, because users are geographically distributed and have no direct contact with, or knowledge of, each other, solution concepts based on common knowledge are not applicable here. We instead must look at the process of learning through repeated play. Traditional approaches to learning through repeated play, which we discuss more fully in Section 5, typically assume the players use their experience to build a model of the likely actions of other players, and then play some form of best response (either exact best response as in the original 'fictitious play' approach (Robinson 1951) or a stochastic best response as in Fudenberg and Levine (1993). Bayesian learning, as in Kalai and Lehrer (1993), is a particular example of this approach, whereby agents begin the game with priors about the expected play of individuals and then update those beliefs as they observe the play. Many of the analyses of such learning algorithms suggest that they result in either Nash or correlated equilibria (see, e.g., Kalai and Lehrer 1993, Fudenberg and Levine 1993, Foster and Vohra 1996).

These results, while important to understanding the rational foundations of equilibria, do not apply in distributed settings, due to the factors we discussed above. In terms of the underlying game users know their own action space, and can observe (after some delay) the payoff resulting from a particular action at a particular time, but do not know their own payoff function, nor any other player's payoff function, and cannot observe the actions of other players. Given this very limited information, users have no sense of what other players are doing, nor any idea of what would constitute a best reply if they did, and so users cannot adopt a fictitious play approach. Instead, we analyze the case in which users (or the software on the machines they are using) employ simple learning algorithms that experiment with various actions and then focus their play on the actions providing the highest payoffs. This is similar in spirit to the 'stimulus-response' approaches studied in Roth and Erev (1995), Borgers and Sarin (1996), and Erev and Roth (1996). Often the work on such learning approaches concentrates on matching the results of the learning algorithm to experimental data. Our focus here is quite different, and has three distinct components.

First, we want to understand the nature of learning in settings like the Internet, where players are geographically distributed and have little or no information about each other and the underlying game. In Section 2 we discuss some of the relevant considerations arising in the Internet and other distributed settings. We then present criteria that all 'reasonable' learning algorithms in this setting must satisfy. The key components are optimization, monotonicity, and responsiveness.

Second, in Section 3, we address the asymptotic result of play among a set of reasonable learners. In a previous paper (Friedman and Shenker 1995), we analyzed one particular family of learning algorithms with these properties. Here, we attempt to identify the class of all learning algorithms that can be considered reasonable, and then study the union of asymptotic plays for all populations of reasonable learners. In other words, if all we know is that agents are reasonable, what predictions can we make about their asymptotic play? We find that the asymptotic play always resides in the serially unoverwhelmed set (defined in Section 3). We are not able to show, and in fact do not believe, that reasonable learning algorithms actually visit (with significantly large probability) all points in the serially unoverwhelmed set. We discuss this in detail in Section 3.5.

Third, in Section 4, we discuss the implications of these convergence results for mechanism design and explore which social choice functions can be implemented in these distributed setting. We find that social choice functions implementable in this decentralized setting must be strictly strategyproof (any deviation that leads to a different outcome results in lower utility for the deviator) and Maskin Monotonic. Moreover, any social choice functions implementable with the serially unoverwhelmed solution concept (which, as stated above is a superset of the true solution concept) must be strictly coalitionally strategyproof (see Section 4 for a definition). We then present examples of some implementable social choice functions.

2 Learning in Distributed Systems

In this section we first informally discuss the nature of learning algorithms appropriate for the Internet. We then formalize these notions of what makes a *reasonable* learning algorithm into precise definitions and provide some examples. It is important to emphasize that we are not claiming that these algorithms are justified by being truly rational or provably optimal in any precise sense. We are merely trying to model the kinds of adaptive learning procedures that either are currently or could potentially be used on the Internet.

2.1 Learning in the Internet

The game-theoretic properties of the Internet are common to many other distributed settings, but for concreteness in the paragraphs below we focus solely on the Internet context. There are four main aspects of the Internet that are particularly relevant to our game theoretic formulation.

First, as discussed above, players typically have extremely limited information. They do not know who the other players are, or even how many, and they do not observe other players' actions. In addition, because they are not aware of the underlying network topology or characteristics, players typically don't know the payoff functions; that is they don't know how their payoffs depend on the actions of other players or even on their own actions. The *only* information available to users are their own actions and the resulting payoffs (and they may only learn the payoff after some delay). This lack of information is actually a central design principle of the Internet. The architectural notion of *layering* (see Tanenbaum 1996 for a textbook discussion of layering) of network protocols is intended to allow computers to utilize the network without knowledge of the underlying physical infrastructure, and to allow applications (such as email, or file transfer) to operate without detailed knowledge of the current level of network congestion.

Second, players do not carry out any sophisticated optimization procedures. Often the actual decisions about resource utilization are made by computer programs (either the application, or lower level protocols like TCP) without direct human intervention. Thus, the "learning algorithm" must be embedded in software, and that limits the flexibility and complexity of the optimization procedure. Moreover, such learning algorithms are intended to be "portable" – i.e., usable on any machine located anywhere – and so are expressly designed to not rely on the details of the specific context.[4] In particular, a Bayesian approach based on updating priors is not realistic here, since the layering of network protocols ensures that any priors would be quite ill-informed. Even in cases where the resource decisions are made directly by the human user, it seems unlikely that the user will be making complex optimization decisions given the very meager information available. Typically the user actions in such cases are limited to adjusting parameter settings for underlying programs (such as adjusting the number of TCP connections a browser opens) rather than actually exercising detailed control.

Third, there is no synchronization and no natural unit of time on the Internet. Players do not all update their actions at the same time (as they do in the standard repeated game literature). To the contrary, the rate at which the updating occurs can vary by several orders of magnitude. Note that there is a delay between when agents update their action and the time they notice a change in their payoffs; for the congestion game described in the Introduction, this delay is typically on the order of a roundtrip time (the time it takes a packet to get to its destination and the acknowledgment[5] to make the return trip). These roundtrip delays vary from 100s of microseconds if the destination is on the same Ethernet (the delay is due to operating system overhead), to 100s of milliseconds if the destination is across the country (the delay is the speed-of-light delay of propagation). Standard control-theoretic results suggest that control loops should not update faster than the roundtrip time.

[4]For example, TCP does not know the content of the data it is conveying, nor does it know anything about the network over which the data is flowing. It merely waits for signs of congestion and responds appropriately.

[5]In TCP, the receipt of each data packet is "acknowledged" by an ACK packet sent from receiver to sender.

Since updating rates are tied to roundtrip times, the variation in update rates will be quite large. Moreover, some learning agents will be people, not programs, and their "updating rates" are most likely on the order of at least seconds, if not significantly slower. Thus the standard model of a repeated game in which players are synchronized can be misleading in the Internet context.[6]

Fourth, and finally, it is neither the long term nor the short term, but the medium term (as defined by Roth and Erev, 1995) that is relevant. Players typically use the system for many time units, measured in their appropriate timescale; however, the nature of their payoff function changes fairly often as new players enter the system, or as the system configuration changes – often due to equipment failures for which the network automatically compensates. The important point here is that players do not know directly that the payoff function has changed; they can only observe the payoffs they get and so can't distinguish between when another player changes her action and when the environment itself changes. This requires the learning algorithm to always be *responsive*, which we define more formally in Section 2.3.

These four properties characterize what we call distributed systems. The natural question, then, is: what forms of learning algorithms are appropriate in distributed systems? We claim that in such settings there are three primary requirements that one would expect of any reasonable learning algorithm. One requirement is that, against a fixed payoff function (i.e., when there are no other players, just nature), the player learns to achieve the optimal payoff. This seems to be the most basic requirement of an optimizing learning algorithm, and it would be hard to justify any algorithm that did not satisfy this criterion. Another reasonable requirement is that the learning algorithm be monotonic in the payoffs; that is, if we modify the payoff function by raising the payoffs for a certain action, the probability of

[6]Lagunoff and Matsui (1995) have also made a similar point about the role of asynchrony in the set of sequential equilibria for repeated games.

the agent playing that action should not decrease. This is similar to the "Law of the Effect" which is well know in the psychology literature, and is discussed by Roth and Erev (1995) as a fundamental property in experimental learning. Finally, many of the learning algorithms in the literature decrease the rate at which they respond with time; in settings like the Internet, where the payoff function changes frequently as players come and go, agents must always be prepared to respond to a new situation in a bounded amount of time. Thus, there are three informal components of being a reasonable learner: optimization, monotonicity, and responsiveness. We now proceed to make these concepts precise, but first we must describe our basic model.

2.2 Model

In this section we describe a simple model to capture the key elements of a distributed setting such as the Internet. Consider a game with a set $\mathcal{P}$ players ($\|\mathcal{P}\| = P$) where each player has a finite action set A_i. The payoffs of the game are described by a time dependent (and possibly stochastic) function $G : A_1 \times A_2 \ldots \times A_P \times \Re \mapsto [0,1]^P$, where for convenience, and to simplify notation, we have restricted payoffs to $[0,1]$.[7] The game is played in continuous time; $a_i(t) \in A_i$ denotes player i's action at time t and $G_i(a(t), t)$ denotes her instantaneous payoff flow at time t. A *stable* game is one in which $G(a,t) = G(a,t')$ for all t, t'; i.e., there is no time dependence. For stable games $< G, A >$ we will drop the last argument from the notation and just write $G_i(a(t))$. Later we will refer to games that are *stable after time* t, which means that $G(a,t'') = G(a,t')$ for all $t'', t' \geq t$.

While the payoffs arise from the game structure, each individual player is completely unaware of the presence of other players and of the payoff function G. Thus, from the perspective of an individual player, we need only model the fact that they receive some

[7] To guarantee that integrals are well defined we assume that on any finite time interval $[s,t]$ the function $G(a,t)$ is continuous in t except at perhaps a finite number of places.

payoff flow $\Gamma_i(t)$. This payoff flow $\Gamma_i(t)$ can depend explicitly on time (perhaps in a stochastic manner) and on all the player's previous actions.

Preferences over different payoff flows can be extremely complex. Here we restrict our attention to a simple case by assuming that players have a fixed sampling rate, evaluating average payoffs at discrete and deterministic epochs.[8] In our model, a player has discrete time horizons $t_i^1, t_i^2, \ldots$ at which she evaluates her payoff as some possibly weighted average of her flow payoffs, and then at the end of the epoch can decide to alter her action.[9] We let $a_i(n)$ be the player i's action chosen at t_i^n which is then maintained until t_i^{n+1}. Note that there is no synchronization in the system, so the time horizons are different for each player; i.e., we can have, and generally do have, $t_i^n \neq t_j^n$ for $i \neq j$. We say that play is *synchronous* if $t_i^n = t_j^n$ for all n and all i, j.

Define

$$\Gamma_i(n) = \int_{t_i^n}^{t_i^{n+1}} G_i(a(t), t) d\beta_i \left(\frac{t - t_i^n}{t_i^{n+1} - t_i^n} \right),$$

where $\beta_i(t)$ is some continuous nondecreasing cumulative distribution function with $\beta_i(0) = 0$, $\beta_i(1) = 1$. Thus $\Gamma_i(n)$ is a weighted average of $\Gamma_i(t)$ over the time period $[t_i^n, t_i^{n+1}]$. Let $h_i^A(n) = (a(1), a(2), \ldots, a(n-1))$, $h_i^\Gamma(n) = (\Gamma_i(1), \Gamma_i(2), \ldots, \Gamma_i(n-1))$ and $h_i(n) = (h_i^A(n), h_i^\Gamma(n))$ be player i's history up to period n, and let $H_i(n)$ be the set of all possible histories for player i. $\Gamma_i(n)$ is a function of the time n, the current action $a_i(n)$, the history $h_i(n)$ and may also be stochastic. For the remainder of this section we will write $\Gamma_i(a_i(n), n, h_i(n))$. In this formulation the other players are modeled as part of the environment; the fact that their behavior is affected by agent i's history of play is incorporated into

[8] Thus, we are not considering anything as complex as the equilibria of repeated games in continuous time (see, for example, Stinchcombe 1992 for a discussion), but are only attempting to analyze the behavior of fairly simple learners.

[9] Note that the decision points are often determined by the technology and are typically not treated as strategic variables. Nonetheless, we believe that most of our results are still valid for learners that strategically manipulate their decision points, given noisy payoffs and delays in observation. In particular, the ability to manipulate decision points should not decrease the set of outcomes that arise.

Γ's dependency on h_i.

Agent i uses a learning algorithm to choose $a_i(n)$. Since, in this setting, agents cannot observe the actions of other agents, their choice of $a_i(n)$ can only depend on the history of agent i's own plays and own payoffs, $h_i(n)$. With such little *a priori* information about the game, players must experiment with various actions in order to learn about the resulting payoffs. Such experimentation is often best done with randomized algorithms. While randomization is often extremely useful, it can be unlucky, and so we must allow for occasional 'mistakes' (i.e., suboptimal behavior). We will consider learning to be sufficiently optimal if it is almost optimal almost all of the time. This type of learning is known as PAC learning – probably approximately correct learning – and can be extremely powerful. See, for example, Valiant (1984) or Blumer et. al. (1989).

Given a payoff function Γ_i and history $h_i(n)$ we must be able to compare the value of different actions. One method, which we choose for its simplicity, is to compare the means of the random variable $\Gamma_i(a_i, n, h_i(n))$. For any $\delta > 0$, we will write $\Gamma_i(a_i, n, h_i(n)) \succ_\delta \Gamma_i(b_i, n, h_i(n))$ to mean that $E[\Gamma_i(a_i, n, h_i(n))] \geq \delta + E[\Gamma(b_i, n, h_i(n))]$.

In the remainder of this section we will consider a single player and thus will drop the subscript i, which will be implicit. Let $\mathcal{E}(N)$ be an environment defined over N periods, i.e. a payoff function defined on $0 \leq n < N$.

2.3 "Reasonable" Learning Algorithms

As we discussed in Section 2.1, the three requirements of a reasonable learner are optimization, monotonicity, and responsiveness. These informal concepts can be made more precise with the help of the following definitions.

The requirement of optimization is simply the notion that, in an environment with a single action that is better – provides higher payoffs – than any other, the learning algorithm should eventually learn to almost always take this optimal action. Certainly one cannot imagine reasonable learning algorithms doing otherwise.

Definition 1 *An environment $\mathcal{E}(N)$ is δ-*simple with optimal action $a^* \in A$ *if, for all $0 \leq n \leq N$,*

$$\Gamma(a^*, n, h(n)) \succ_\delta \Gamma(a, n, h(n))$$

for all $a \in A$ such that $a \neq a^*$, for all $h(n) \in H(n)$.

A reasonable learner should be able to learn the optimal action in such games if N is sufficiently large. A learning algorithm or learner, L, is a mapping from histories $h(n)$ to probability distributions over actions in A. Given an environment $\mathcal{E}(N)$ this induces a probability distribution over the set of all histories, $H(n)$, which we will denote $\mu_{L,\mathcal{E}(N)}$.

Definition 2 (Optimization) *A player is a* simple $(\epsilon, \delta, N, \omega)$ learner *if, for any $\mathcal{E}(N')$ which is δ-simple with optimal action $a^* \in A$, such that $N' \geq N$, and any m such that $N \leq m \leq N'$, there exists a subset $\hat{H}(m) \subseteq H(m)$ such that $\mu_{L,\mathcal{E}(N)}(\hat{H}(m)) > 1 - \omega$, and for all $h(m) \in \hat{H}(m)$, $Pr[a(m) = a^* \mid h(m)] > (1 - \epsilon)$.*

Simple learners can find the optimal action in simple games, in the sense of playing the optimal action with high probability for "most" histories, where "most" is defined by the probability distribution induced by the learner. Note that the probabilistic formulation of the above definition, with the allowance of occasional 'mistakes', is necessary since a randomized learning algorithm can be 'unlucky.'

Now we attempt to capture the more general idea of responsiveness, or medium term learning. Let $H_x(m)$ denote the set of all histories on $[x, m)$.

Definition 3 (Responsiveness) *A learner is* $(\epsilon, \delta, N, \omega)$-responsive *if, given any environment $\mathcal{E}(N')$ and any $N \leq m \leq N'$ such that $\mathcal{E}(N')$ restricted to $[m - N, m]$ is δ-simple* *with optimal action a^* there exists (on $\mathcal{E}(N')$ restricted to $[m-N,m]$) a subset $\hat{H}_{m-N}(m) \subseteq H_{m-N}(m)$ such that $\mu_{L,\mathcal{E}(N')}(\hat{H}_{m-N}(m)) > 1-\omega$, and for all $h(m) \in \hat{H}_{m-N}(m)$, $Pr[a(m) = a^* \mid h(m)] > (1-\epsilon)$.*

Being $(\epsilon, \delta, N, \omega)$-responsive requires that the learner respond to changes in the environment within a bounded time, N; that is, in any period of length N during which the environment has been δ-simple, the learning algorithm must converge (in a probabilistic sense) to the optimal action.[10]

Note that responsiveness is strictly stronger than being a simple learner. For example, consider the following "quasi-static" environment in which every τ periods the optimal action may change, but in between changes the environment is δ-simple. Let $I(n)$ be the indicator variable which is 1 when the agent chooses the optimal action in time period n and 0 otherwise. We consider the case where these stable intervals can vary, and so we can let τ be a random variable with mean $\overline{\tau}$.

Theorem 1 *In the quasi-static environment*

$$\lim_{\overline{\tau}\to\infty} \lim_{m\to\infty} \frac{1}{m\overline{\tau}} \sum_{t=0}^{m\overline{\tau}} I(t) \geq (1-\epsilon)(1-\omega),$$

almost surely, for any $(\epsilon, \delta, N, \omega)$ responsive learner.

Proof: Let $\overline{\tau} = rN$, for $r > 4$, and consider a period of length τ where the environment is δ-simple. With probability greater than $1-1/\sqrt{r}$ the period is longer than $\sqrt{r}N$. Then, for that period $E[I(T)] > (1-\epsilon)(1-\omega)(r-\sqrt{r})/r$. Note that this bound is independent of all previous periods and since with probability $1-1/\sqrt{r}$ the bound holds, we get $\lim_{m\to\infty} \frac{1}{m\overline{\tau}} \sum_{t=0}^{m\overline{\tau}} I(t) >$

[10] Most adaptive learning algorithms in the literature (Fudenberg and Levine 1993, Erev and Roth 1996, Borgers and Sarin 1995) are not adaptive, because as time goes on they become less reactive to changes in their environment. In theory, Bayesian-type learners (e.g. Kalai and Lehrer 1995, Foster and Vohra 1996) could satisfy responsiveness by including the possibility of switching in the priors. Because the space of all possible environmental changes is huge, and players are ill-informed about their probabilities, this would result in an algorithm that is extremely difficult to implement and completely impractical.

$(1-\epsilon)(1-\omega)(1-1/\sqrt{r})(r-\sqrt{r})/r$, almost surely, and taking the limit as $r \to \infty$ completes the proof. $\Box$

Note that nonresponsive learners do not satisfy this theorem. For example, "no regret" learners such as those in Foster and Vohra (1997) do quite badly; $\lim_{\overline{\tau}\to\infty} \lim_{m\to\infty} \frac{1}{m\overline{\tau}} \sum_{t=0}^{m\overline{\tau}} I(t)$ can be on the order of $1/|A|$.[11]

Our next definitions formalize a notion of monotonicity or the "Law of the Effect" (Thorndyke 1898). First we define what it means for one history to be better with respect to an action.

Definition 4 *Given two histories $h(n)$ and $\tilde{h}(n)$ we say that $h(n)$ is* higher *with respect to action $a \in A$ if $h^A(n) = \tilde{h}^A(n)$, and $(h^\Gamma(n))_m \leq (\tilde{h}^\Gamma(n))_m$ whenever $(h^A(n))_m \neq a$ and $(h^\Gamma(n))_m \geq (\tilde{h}^\Gamma(n))_m$ whenever $(h^A(n))_m = a$.*

Definition 5 (Monotonicity) *A learner is* monotonic *if for any pair of histories $h(n), \tilde{h}(n)$ such that $h(n)$ is higher with respect to $a \in A$ than $\tilde{h}(n)$, then*

$$Prob[a(n) = a \mid h(n)] \geq Prob[a(n) = a \mid \tilde{h}(n)]$$

Combining these definitions, we can now precisely define what we consider to be a reasonable learning algorithm in distributed settings like the Internet.

Definition 6 *A learner is an $(\epsilon, \delta, N, \omega)$ reasonable learner if it is monotonic and $(\epsilon, \delta, N, \omega)$ responsive.*

Note that monotonicity allows us to make statements about environments that are not "simple". For example, in an environment there may be several actions, any one of which may be optimal depending on exogenous effects, but there may also be actions that are clearly suboptimal. In this case we can show that such clearly suboptimal actions will be played rarely by a reasonable learner.

[11] This is demonstrated numerically in Greenwald, Friedman and Shenker (1998).

Theorem 2 *Consider an environment $\mathcal{E}(N')$. Assume that there is an action $a^* \in A$ and a set of actions $\tilde{A} \subset A$ such that all actions in $\tilde{A}$ are always worse than a^*, $\Gamma(a^*, n, h(n)) \succ_\delta \Gamma(a, n, h(n))$ for all $a \in \tilde{A}$. If a player is a $(\epsilon, \delta, N, \omega)$ reasonable learner with $N' > N$ then for any m with $N \leq m \leq N'$ there exists a subset $\hat{H}(m) \subseteq H(m)$ such that $\mu_{L,\mathcal{E}(N')}(\hat{H}(m)) > 1 - \omega$, and for all $h(m) \in \hat{H}(m)$, $Pr[a(m) \in \tilde{A} \mid h(m)] < \epsilon$.*

Proof: Consider the environment in which has the same payoffs as $\mathcal{E}(N')$ when either the action $a = a^*$ or $a \in \tilde{A}$, but has zero payoff for any other action. This environment is δ-simple with optimal action a^*, and thus $Pr[a(m) \in \tilde{A} \mid h(m)] < \epsilon$ by Theorem 1. However for all $a \in \tilde{A}$ this environment is higher than $\mathcal{E}(N')$. Thus in $\mathcal{E}(N')$, the probability of playing $a \in \tilde{A}$ can not be larger than this. $\square$

2.4 Examples

Each of the three notions – optimizing, monotonicity, and responsiveness – that comprise our definition of reasonableness seem, on the surface, to be quite natural and undemanding requirements. Surprisingly, few formal learning algorithms in the economics literature satisfy this definition of reasonableness.[12] Many of the learning algorithms in the standard literature do not have the responsive property; typically their responsiveness to changes in payoffs, or their level of experimentation, diminishes over time. We also note that there are no deterministic algorithms which are responsive.[13]

We now present two examples of *reasonable* learning algorithms.

2.4.1 Stage Learners

The first is a "stage learner", which is a very simple reasonable learner. The stage learner SL_ϵ learns in "stages" of length $1/\epsilon^3$. During each stage, the action that had the highest

[12]With suitable choices of parameters Roth and Erev's (1995) model of learning is "reasonable."
[13]A slight variant of this statement is proven in Fudenberg and Levine (1995).

average in the previous stage (with ties broken randomly) is played with probability $1-\epsilon$, while the remaining actions are each played with probability $\epsilon/(|A|-1)$. The choice of action in any time period is i.i.d. Note that the stage learner almost always plays the action with highest expected value (based on the payoffs observed in the last stage), but 'experiments' with sufficient frequency to notice changes in the environment and react to them.

Theorem 3 *For sufficiently small $\epsilon > 0$ SL_ϵ is an $(\epsilon, \sqrt{\epsilon}, 2/\epsilon^3, (|A|+1)\exp(-(\sqrt{|A|}\epsilon)^{-1}))$ reasonable learner.*

Proof: Assume that during a particular period of length $2/\epsilon^3$ the environment is δ-simple with optimal action a and $\delta = \sqrt{\epsilon}$. Then the stage learner will have faced a δ-simple environment during its previous stage. Define $\hat{\Gamma}(a', n, h(n)) = \Gamma(a', n, h(n)) - E[\Gamma(a, n, h(n))]$. Note that not restricting $\hat{\Gamma}$ to $[0, 1]$ does not affect the stage learner. In this environment $E[\hat{\Gamma}(a, n, h(n))] = 0$ and $E[\hat{\Gamma}(a', n, h(n))] \leq -\sqrt{\epsilon}$ for all $a' \neq a$. Note that $Var[\hat{\Gamma}(a', n, h(n))] \leq 1$ for all $a' \in A$, since $\Gamma(\cdot) \in [0, 1]$.

Define a stage to be 'normal' if each action has been played at least $(2|A|\epsilon^2)^{-1}$ times. The expected number of plays for any particular action is greater than $(|A|\epsilon^2)^{-1}$ while the standard deviation (of the number of times it is played) is less than $(\sqrt{2|A|\epsilon^2})^{-1}$. Thus, from the central limit theorem, the probability of an action not being played at least $(2|A|\epsilon^2)^{-1}$ times is less than $erf((\sqrt{2|A|\epsilon^2})^{-1})$ which is bounded by $\exp(-(\sqrt{|A|}\epsilon)^{-1})$ so the probability of a stage being normal is greater than

$$(1 - \exp(-(\sqrt{|A|}\epsilon)^{-1}))^{|A|} = \sum_{j=0}^{|A|}(-1)^j \exp(-j/(\sqrt{|A|}\epsilon))\frac{|A|!}{j!(|A|-j)!)}$$

When $|A|$ is odd we can rearrange terms to get

$$1-|A|\exp(-j/(\sqrt{|A|}\epsilon))+\sum_{i=1}^{|A|/2}\exp(-2j/(\sqrt{|A|}\epsilon))\frac{|A|!}{2j!(|A|-2j)!)}\left(1-\exp(-1/(\sqrt{|A|}\epsilon))\frac{|A|-2j}{2j+1}\right)$$

$$> 1 - |A|\exp(-j/(\sqrt{|A|}\epsilon))$$

since the terms in the sum are all positive for sufficiently small $\epsilon > 0$. When $|A|$ is even we use the same argument after noting that $(1-\exp(-(\sqrt{|A|}\epsilon)^{-1}))^{|A|} > (1-\exp(-(\sqrt{|A|}\epsilon)^{-1}))^{|A|+1}$.

Define $\gamma(a')$ to be the average payoff for action $a' \in A$ over a normal learning stage. The standard deviation of $\gamma(a')$ is less than $\sqrt{2|A|\epsilon^2}$ while the average is 0 if a' is the optimal action and less than $-\sqrt{\epsilon}$ if a' is not optimal. Thus, the probability of the optimal action having average less than $-\sqrt{\epsilon}/2$ is less than $\exp(-1/\sqrt{|A|}\epsilon)$ since the sequence $\hat{\Gamma}$ is a martingale. (See, Hoeffding, 1994 for details.) This is also the probability of a nonoptimal action having payoff greater than $-\sqrt{\epsilon}/2$. Thus, the probability of the optimal action having the highest payoff is greater than $(1-\exp(-2/\sqrt{|A|}\epsilon))^{|A|}$ which is greater than $1-(|A|+1)\exp(-j/(\sqrt{|A|}\epsilon))$, completing the proof. □

Note that if there are two optimal actions, then the stage learner will alternate randomly between them. For constructive purposes it is often useful to make this choice deterministic. Let $\hat{\Sigma}(A)$ be the set of all strict orderings on A, e.g., for $\rho \in \hat{\Sigma}(A)$, $\rho = (\rho(1), \ldots, \rho(|A|))$ with $\rho(i) \in A$ and $\bigcup_{i \in A} \rho(i) = A$. Then, given an ordering $\rho \in \hat{\Sigma}(A)$, define the ρ-prioritized stage learner, SL^{ρ}_{ϵ}, to be a stage learner that plays, with probability $(1-\epsilon)$, the highest ranking (according to ρ) strategy whose average payoff in the last stage was no less then $\delta/2$ less than the average payoff from any other strategy; remaining actions are still played with probability $\epsilon/(|A|-1)$.

Note that modification of the stage learner has no effect for a $\delta = \sqrt{\epsilon}$-simple environment other than slightly increasing the probability that the learner mistakes the action with the highest payoff.

Theorem 4 *For sufficiently small $\epsilon > 0$ and any $\rho \in \hat{\Sigma}(A)$, SL^{ρ}_{ϵ} is an $(\epsilon, \sqrt{\epsilon}, 2/\epsilon^3, (|A|+1)\exp(-(2\sqrt{|A|}\epsilon)^{-1}))$ reasonable learner.*

Proof: The proof is identical to the previous proof for ordinary stage learners, except for the conditions under which it chooses the "incorrect" optimal action. This may arise when the average payoff for a suboptimal action is within $\delta/2$ of the average payoff for the optimal action, which changes the probability of a mistake slightly. □

2.4.2 Responsive Learning Automata

Our second example is the responsive learning automata (RLA) which was studied in Friedman and Shenker (1995) and motivated the analysis in this paper. RLAs are based on algorithms studied in the engineering literature and have been implemented for many network optimization tasks (see e.g., Chrysalis and Mars 1981, Mason and Gu 1986, and Shrikantakumar 1986). They are also closely related to several models proposed for experimental economic learning (Arthur 1991, Mookerji and Sopher 1996, Roth and Erev 1995). An RLA consists of a probability vector, which can be interpreted as a mixed action at every decision epoch – with probability $p_a(n)$ action a is played. After action a is played and the payoff $\Gamma(n)$ is observed, the probability vector $p_a(n+1)$ is updated by the following rule.

$$p_a(n+1) = p_a(n) + \epsilon^2 \Gamma(n) \sum_{b \neq a} c_b(n) p_b(n)$$

$$\forall b \neq a \;\; p_b(n+1) = p_b(n) - \epsilon^2 \Gamma(n) c_b(n) p_b(n)$$

where

$$c_b(n) = \min[1, \frac{p_b(n) - \epsilon^2/2}{\epsilon^2 p_b(n) \Gamma(n)}].$$

We will denote these learners by RLA_ϵ.

Theorem 5 *For $\epsilon > 0$ sufficiently small, there exist constants $\alpha, \beta > 0$ such that RLA_ϵ is an $(\epsilon, \epsilon, 1/\epsilon^3, \alpha \exp(-\beta/\epsilon))$ reasonable learner.*

Proof: This follows directly from Friedman and Shenker (1995) Theorem 1.

3 Groups of Reasonable Learners

3.1 Context and Definitions

Our discussion of learning algorithms considered an environment seen by a single player which consisted of a general payoff function Γ, with no restriction on how these payoffs were generated. Here we return to the original situation where this payoff function arises from a game G involving P players (with $\mathcal{P}$ denoting the set of players), each with action space A_i.

When focusing on a single player in a general environment, results like Theorem 2 allow us to make some statements about the asymptotic nature of play of a reasonable learner as defined in Section 2.3. Similarly, in this section we assume that each of the P players is a reasonable learner, and ask what the asymptotic nature of the joint play is. This asymptotic set of actions is the *solution concept* appropriate for learning in distributed systems like the Internet. Note that the solution concept must contain the eventual play of all possible sets of learning algorithms. We are not interested in results for one particular learning algorithm, even if the set of such learners have particularly nice convergence properties. All we can assume is that learners are reasonable, not that they conform to some specific algorithm.

Milgrom and Roberts (1990) define an "adaptive learner" as one who eventually eliminates actions that are strictly dominated (in pure actions) over time. They prove that when a group of adaptive learners play together they converge to the serially undominated set (the result of the iterated deletion of these dominated actions).

In this section we parallel those results with two main distinctions. First, we only assume that players are reasonable learners, as defined in the previous section. In this setting it is *not* true that players always eventually abandon dominated actions. Players cannot explicitly identify dominated actions (because they don't know the payoff matrix) and furthermore we show that in some cases dominated actions can even be played in equilibrium. Thus, we can only impose the requirement of reasonableness (as we have defined it) on learners. Second, since in this distributed setting no action can ever be completely discarded, the convergence to any set of actions (or the elimination of others) is only approximate.[14] The fact that all actions remain in play forever makes the analysis of the joint play quite delicate.

As we shall see, a set of reasonable learners need not converge to the serially undominated set. The main result of this section is that a set of reasonable learners eventually play in the serially unoverwhelmed set, the set remaining after iterated elimination of overwhelmed actions. We do not believe this characterization is tight, in that there are some games where no set of reasonable learners will eventually play (with significant probability) in some portions of the serially unoverwhelmed set. However, the serially unoverwhelmed solution concept is the tightest "local set based" solution concept possible, where local set based solution concepts are the natural generalizations of the serially undominated set. Moreover, we present another two sets, the Stackelberg correlated set and the Stackelberg undominated set, and raise the question as to whether the true solution concept lies between these two.

Before proceeding, we require two definitions.

Definition 7 *A "local" dominance operator on a stable game $< G, A >$ is a set of monotone operators, one for each i, $\Lambda_i^{<G_i,A>} : 2^{A_{-i}} \rightarrow 2^{A_i}$ (the notation represents the fact that Λ_i only depends on player i's payoff matrix G_i). We denote this set of operators by $\Lambda^{<G,A>}(\cdot)$ where $(\Lambda^{<G,A>}(\alpha))_i = \Lambda_i^{<G_i,A>}(\alpha_{-i})$ for each i and $\alpha \subseteq A$. (Note that an operator is monotone if for $\alpha, \beta \in 2^{A_{-i}}$ such that if $\alpha \subseteq \beta$, then $\Lambda_i^{<G_i,A>}(\alpha) \subseteq \Lambda_i^{<G_i,A>}(\beta)$.)*

Each local dominance operator describes the set of possible strategies agent i might employ as a function of the possible plays the other agents might make.[15] For each local

[14]Recall that a reasonable learner, in order to remain responsive, can never completely stop playing an action since exogenous effects could modify the payoffs making that action optimal at some later time.

[15]Duggan and Le Breton (1997) study the fixed points of local dominance operators, which they denote "Dominance Structures."

dominance operator $\Lambda^{<G,A>}(\cdot)$ we can define the related solution concept.

Definition 8 *Given a local dominance operator* $\Lambda^{<G,A>}(\alpha)$*, the associated* local set based solution concept *(LSB) is the operator* Λ^∞ *defined by* $\Lambda^\infty(G,A) = \lim_{m\to\infty}(\Lambda^{<G,A>})^m(A)$.[16]

One standard LSB is defined using dominated actions. The local dominance operator is given by $\Lambda_i^{<G_i,A>}(\alpha) = \{a_i \in A_i \mid \nexists b_i \in A_i \;\; s.t. \;\; \forall a_{-i} \in \alpha \quad G_i(a) < G_i(b_i, a_{-i})\}$. We will denote the LSB for this operator by D^∞, and so $D^\infty(G,A)$ denotes the serially undominated set of the game $< G, A >$.

The relevant LSB for decentralized games is based on unoverwhelmed actions. The local dominance operator is

$$\Lambda_i^{<G_i,A>}(\alpha) = \{a_i \in A_i \mid \nexists b_i \in A_i \;\; s.t. \;\; \forall a_{-i}, b_{-i} \in \alpha \quad G_i(a) < G_i(b)\}.$$

We will denote the LSB that results from the iteration of this operator by O^∞, and refer to $O^\infty(G,A)$ as the serially unoverwhelmed set of the game $< G, A >$. (We will occasionally abbreviate this as $O^\infty(G)$ when the action subset is the entire action set, and will further abbreviate the notation to O^∞ when the game is also unambiguous. Similarly, when the game is unambiguous and the action subset is the entire action set, we will use the notation O^k to denote the k'th iteration of the unoverwhelmed local dominance operator applied to the entire action set.)

For comparison, note that one action dominates another if all payoffs for the one are greater than the other for all given *fixed* sets of other players' actions. In contrast, one action overwhelms another if all payoffs, over all sets of other players' actions, for the one are greater than all payoffs , over all sets of other players' actions, for the other. Domination compares the 'vector' of payoffs term-by-term; overwhelming compares the entire 'bag' of payoffs available, and thus is a much stronger requirement.

[16]The limit exists since $\Lambda^{<G,A>}$ is a monotone set operator, and A is finite.

For any game, the serially unoverwhelmed set contains the serially undominated set, which contains the set of rationalizable actions.

3.2 Convergence Results

Given a finite set of reasonable learners $L = \{L_1, \ldots, L_m\}$ where each L_i is an $(\epsilon_i, \delta_i, N_i, \omega_i)$ reasonable learner, let $(\epsilon, \delta, N, \omega)(L) = (\max_i \epsilon_i, \max_i \delta_i, \max_i N_i, \max_i \omega_i)$. Now consider a repeated game played by these players with payoff functions $G_i(a(t), t)$ and let τ_i^+ be the largest time interval between player i's decision epochs, τ_i^- the smallest, and let $\tau^+(L) = \max_i \tau_i^+$ and $\tau^-(L) = \min_i \tau_i^-$. Define $\alpha(L) = N(L)\frac{\tau^+(L)}{\tau^-(L)}$ and let $|A| = \Pi |A_i|$.

Note that a set of learners L and a game $< G, A >$ induce a measure over histories H by their play, which we will call $\mu_{L,G}$. We now present our main result which is that decentralized learning leads to the serially unoverwhelmed set.

Theorem 6 *Given any game $G(a(\cdot), \cdot)$ which is stable after time t and any $\hat{\omega} > 0$. There exists $(\epsilon', \delta', N', \omega') > 0$ such that for any $s > t + N'$, and any set L of reasonable learners playing satisfying $(\epsilon, \delta, N, \omega)(L) \leq (\epsilon', \delta', N', \omega')$, and $\omega(L)\alpha(L) \leq \hat{\omega}$ the players "converge" to O^∞ in the following sense: there exists a set $\hat{H}(s) \subseteq H(s)$ with $\mu_{L,G}(\hat{H}(s)) > 1 - \omega$ such that $Pr[a(s) \in O^\infty(G(\cdot, s)) \mid h(s)] > 1 - \epsilon'$.*

Proof: Fixing a game $< G, A >$, choose an action $a_i \notin O_i^k$ and define

$$\beta_{k,i}(a_i) = \max_{b_i \in O_i^k} \left([\min_{b_{-i} \in O_{-i}^{k-1}} G_i(b_i, b_{-i})] - [\max_{a_{-i} \in O_{-i}^{k-1}} G_i(a_i, a_{-i})] \right),$$

and let $\beta_{k,i} = \min\{\beta_{k,i}(a_i) \mid a_i \notin O_i^k\}$ and note that $\beta_{k,i} > 0$ if an action is eliminated for player i at round k, and otherwise $\beta_{k,i} = 0$. Let $\beta = \max_{k,i} \beta_{k,i}$.

Define time interval k by $I_k = [t + k\tau^+(L)N(L), (t + k + 1)\tau^+(L)N(L)]$. Note that in I_1 all play is in O^0. We proceed inductively. Assume that for any s in period I_k learner i is playing in O^k with probability greater than $1 - \epsilon(L)$. If $m\epsilon(L) + \delta_i \leq \beta$ then learner i is playing in an environment in which all actions not in O^{k+1} must be exceeded in expected value by those actions in O^{k+1} and thus we can apply Theorem 2 to show that the learner learns to play these actions in period $k+1$ with probability less than ϵ_i with probability greater than $1-\omega_i$. The probability that the player does this at every interval in period $k+1$ is greater than $1-\alpha(L)\omega(L)$. Thus the probability that all learners do this is greater than $1-m\alpha(L)\omega(L)$. Finally, the probability that this occurs over all stages is greater than $1-m|A|\alpha(L)\omega(L)$, since there can be at most $|A|$ stages required to reach O^∞. Thus if $\omega' > m|A|\alpha(L)\omega(L)$ this shows that convergence will occur. □

This theorem immediately applies to Stage Learners and RLAs.

Corollary 1 *There exists some $\epsilon', p > 0$ such that any group L of Stage learners and RLAs satisfying $\epsilon(L) < \epsilon'$ and $(\max_i \epsilon_i)^p < \min_i \epsilon_i$ converge to the serially unoverwhelmed set, where convergence is defined as in Theorem 6.*

The above results hold for a stable games. However, the analogous results hold even with time-varying games. For instance, consider (as we did in Section 2.3 for games against nature) the "quasi-static" game in which every τ periods the payoff functions may change, but in between changes the game is constant. Let $I(t)$ be the indicator variable which is 1 when current action is in the serially unoverwhelmed set, $a(t) \in O^\infty(G(\cdot, t))$. Let τ be a random variable with mean $\overline{\tau}$. Then the Theorem also implies convergence in this game.

Corollary 2 *In the quasi-static game just described*

$$\lim_{\tau\to\infty} \lim_{m\to\infty} \frac{1}{m\tau} \sum_{t=0}^{m\tau} I(t) \geq (1-\epsilon(L))(1-\omega(L)),$$

for any group of learners satisfying the conditions in the previous theorem.

As discussed in Section 2.3, nonresponsive learners, including no regret learners, do very poorly in quasi-static environments.

3.3 Synchronous Play

Interestingly, if we restrict to sets of players who play synchronously, $t_i^n = t_j^n$ for all i, j, n, then we revert to the standard results – play converges to $D^\infty(G)$.

Theorem 7 *Let L be a set of reasonable learners playing* synchronously *a game $G(a(\cdot), \cdot)$ which is stable after time t. Then for any $\hat{\omega} > 0$ there exists $(\epsilon', \delta', N', \omega') > 0$ such that for any $s > t + N'$, if $(\epsilon, \delta, N, \omega)(L) \leq (\epsilon', \delta', N', \omega')$, and $\omega(L)\alpha(L) \leq \hat{\omega}$ then there exists a set $\hat{H}(s) \subseteq H(s)$ with $\mu_{L,G}(\hat{H}(s)) > 1 - \omega$ such that $Pr[a(s) \in D^\infty(G(\cdot, s)) \mid h(s)] > 1 - \epsilon'$.*

Proof: The proof of this theorem is analogous to Theorem 6 after noting that in a synchronous game the expected payoff of any dominated action is always less than that of the dominating action, since for player i, a_{-i}^t is governed by a random distribution that is does not depend on the choice of a_i^t (although they may be correlated ex post). □

We do not know if there is a smaller set (e.g., the support of the set of correlated equilibria, or the rationalizable strategies) for which this result continues to hold.

3.4 Minimal Solution Concepts

Theorem 6 establishes a bounding set on the asymptotic play. The true solution concept may be somewhat smaller. Let $\mathcal{C}(G, A) \subseteq A$ be the true solution concept; that is, the union of the set of strategies played with non-negligible probability by all possible groups of reasonable learners. More formally, we have the following definition.

Definition 9 *$\mathcal{C}(G, A)$ is the smallest set for which Theorem 6 is true when O^∞ is replaced by $\mathcal{C}$.*

First we will show that $\mathcal{C}(G, A)$ contains some Stackelberg equilibria. Given a strict order σ on $\mathcal{P}$, $\sigma \in \hat{\Sigma}(\mathcal{P})$, define the Stackelberg game G_σ to be the (extensive form) game with payoffs given by G in which player $\sigma(1)$ moves first, then $\sigma(2)$, and continuing up to player $\sigma(P)$.

Definition 10 *A Stackelberg equilibrium with respect to order $\sigma \in \hat{\Sigma}(\mathcal{P})$ is a subgame perfect equilibrium of the Stackelberg game G_σ.*

The key aspect of the following proof is the observation that by separating their timescales, players behave as if they are playing a Stackelberg game. Note that the role of leader is not intentional by the learner – in fact the learner is not even aware that it is the leader – and is merely the product of learning slowly. Thus, learning slowly, usually perceived as a disadvantage, provides the benefits of being a Stackelberg leader. This is an example where superior sophistication (such as faster computer processors or better learning algorithms) may lead to inferior results.

Theorem 8 *For every ordering $\sigma \in \hat{\Sigma}(\mathcal{P})$ there exists some $a \in \mathcal{C}(G, A)$ such that a is a Stackelberg equilibria for the game $< G, A >$ with respect to order σ.*

Proof: Consider a group of identical prioritized stage learners, $SL_\epsilon^{\rho^j}$, each with any ordering $\rho^j \in \hat{\Sigma}(A_j)$. Choose ϵ such that

$$\sqrt{\epsilon} \leq \min_{a,a' \in A,\ i \in \mathcal{P}} \{|G_i(a) - G_i(a')| \quad : G_i(a) \neq G_i(a')\}.$$

Set $t^n_{\sigma(i)} = n\tau_i$, where $\tau_i = \lceil 4/\epsilon^3 \rceil^{P-i}$ (where $\lceil x \rceil$ is the least integer greater than x), so the players update at fixed, but different, intervals with the first player in the Stackelberg ordering being the slowest. Lastly, let $\beta(t) = 0$ for $t \in [0, 1/2]$ and $\beta(t) = 2(t - 1/2)$ for $t \in (1/2, 1]$, so players average over payoffs only during the second half of their time interval.

Now consider player $\sigma(P-1)$. She chooses an action at time $t^n_{\sigma(P-1)}$, and in the (open) interval between $t^n_{\sigma(P-1)}$ and $t^{n+1}_{\sigma(P-1)}$ no player before her in the order will change their current action and player $\sigma(P)$ will converge to a best reply (with high probability) to the current action $a_{-\sigma(P)}$ by time $t^n_{\sigma(P-1)} + \tau_{P-1}/2$. Thus, from her point of view, the game is a Stackelberg one, where P follows her, since she only evaluates payoffs between $t^n_{\sigma(P-1)}+\tau_{P-1}/2$ and $t^{n+1}_{\sigma(P-1)}$. Continuing backwards through the ordering we see that each player follows the player before her, and that play will converge to the specified equilibria. □

Lastly, Theorem 6 shows that $\mathcal{C}(G,A) \subseteq O^\infty(G,A)$. As we discuss later, we suspect that this inequality is strict for some games. However, if we restrict ourselves to LSBs, then Theorem 6 is tight in the following sense.

Theorem 9 *Let Λ be an LSB such that $\Lambda^{<G,A>}(\alpha) \subseteq O^{<G,A>}(\alpha)$ for all $< G,A >$ and $\alpha \subseteq A$, and $\mathcal{C}(G,A) \subseteq \Lambda^\infty(G,A)$ for all $< G,A >$. Then, $O^\infty(G,A) = \Lambda^\infty(G,A)$ for all $< G,A >$.*

Proof: Assume that there exists a game $< G,A >$ such that $O^\infty(G,A) \supset \Lambda^\infty(G,A)$. Thus, there must exist some k such that $\Lambda^{k+1}(A) \neq O^{k+1}(A)$ but $\Lambda^k(A) = O^k(A)$, where we drop the superscript $< G,A >$ here and below for notational convenience. Let $\alpha = \Lambda^k(A) = O^k(A)$. Choose some $b_i \in A_i$ that does appear in $O_i(\alpha_{-i})$ but not in $\Lambda_i(\alpha_{-i})$. Now construct the game $< G',A >$ in the following manner. For all $a \in A$ set $G_i(a) = G'_i(a)$.

Choose a function $r : A_i \to A_{-i}$ such that for all $a_i \neq b_i$

$$r(a_i) \in \mathrm{argmin}_{a_{-i} \in \alpha_{-i}} G_i(a_i, a_{-i})$$

and

$$r(b_i) \in \mathrm{argmax}_{a_{-i} \in \alpha_{-i}} G_i(b_i, a_{-i}).$$

Now for all $j \neq i$ define $G'_j(a_i, a_{-i}) = 1$ when $a_{-i} = r(a_i)$, $G'_j(a_i, a_{-i}) = 0$ when $a_{-i} \notin \alpha_{-i}$, and $G'_j(a_i, a_{-i}) = 1/2$ for all other cases.

By construction $O^{<G',A>}(A)_{-i} = \alpha_{-i}$ and thus $b_i \notin \Lambda^{<G',A>}(O^{<G',A>}(A))_i$, since $G_i = G'_i$. Since $\Lambda^{<G',A>}(A) \subseteq O^{<G',A>}(A)$ this implies that $b_i \notin \Lambda^{<G',A>}(\Lambda^{<G',A>}(A))_i$ by monotonicity which also implies that $b_i \notin \Lambda^\infty(G', A)$. However, we will now show that $b_i \in \mathcal{C}(G', A)$ proving the theorem.

Construct a Stackelberg ordering where player i is the leader, i.e., first in the ordering, and let her be a prioritized stage learner where action b_i is the top priority action. Let the other players be in any order and assume that they are ordinary stage learners.

Now we use the same construction as in the previous theorem to show that the outcome of this game is the strategy profile $(b_i, r(b_i))$ since all followers will play $r_j(a_i)$ in response to the leader's action and the leader will then see a game in which action b_i has the highest payoff, by construction of the function $r(\cdot)$ and the fact that it is not overwhelmed. Note that this payoff may not be strictly highest; however the action b_i will be chosen because of the priority ordering used. □

3.5 A Tighter Solution Concept?

While the O^∞ solution concept is the tightest LSB solution concept, it is probably not the tightest solution concept for decentralized learning. That is, we expect that there are games for which $\mathcal{C}(G, A) \neq O^\infty(G, A)$Consider the following game:

	L	R
T	1,1	.3,.6
B	.6,.3	0,0

O^∞ of this game is the set of all actions. It seems intuitive, although we have no formal proof, that any pair of decentralized learners will converge to (T, L). In Greenwald, Friedman and Shenker (1998) simulations of the RLAs and Stage Learners were consistent with this intuition. Since our goal here is to describe the possible outcomes of a game played by decentralized learners it is important to find the tightest solution concept to which decentralized learners converge.

We now describe a class of solution concepts which is suggested by the proof of Theorem

9. We do not know whether any of these is the "correct" solution concept; we introduce these solution concepts to formulate a testable open question, whose resolution would greatly improve our understanding of reasonable learners in distributed settings.

3.5.1 Stackelberg Solution Concepts

Consider some solution concept $\Upsilon(G, A)$ that is deemed appropriate for synchronous games. We now define a solution concept based on $\Upsilon(G, A)$ that more appropriate for games with arbitrary degrees of asynchrony. Given a (finite) set of players $\mathcal{P}$ (with $P = |\mathcal{P}|$), define a (non strict) play order $\sigma = \sigma(1), \ldots, \sigma(m)$ where $\sigma(r) \subseteq \mathcal{P}$, $\bigcup_{r \in \{1,\ldots,m\}} \sigma(r) = \mathcal{P}$, and for $r \neq r'$, $\sigma(r) \bigcap \sigma(r') = \emptyset$. Let $\Sigma(\mathcal{P})$ be the set of all (non strict) play orders, $\sigma^{-}(r) = \bigcup_{r' \in \{1,\ldots,r-1\}} \sigma(r')$, and $\sigma^{+}(r) = \bigcup_{r' \in \{r+1,\ldots,m\}} \sigma(r')$.

Given a play order σ define the associated Stackelberg game where players move according to that order. Each player takes the actions of the players earlier in the order as a given and plays accordingly. Thus, a player sees the behavior of the earlier players as fixed, and sees the later players as reacting to their moves. Each player's elemental action in this Stackelberg game is actually a *response function*, in which an action of the underlying normal form game a_i is chosen as a function of the actions of the previous (in terms of the ordering) players. That is, for agent $i \in \sigma(r)$, a strategy in the Stackelberg game is a response function[17] $\phi_i : A_{\sigma^{-}(r)} \mapsto A_i$. Let $\mathcal{G}^{\sigma}$ be the set of all such Stackelberg strategies ϕ for the ordering σ, and let $\mathcal{G}^{\sigma,r}(\sigma)$ be the restriction of $\mathcal{G}^{\sigma}$ to $\sigma^{+}(r)$. For $\phi \in \mathcal{G}^{\sigma}$, let $Out_i^{\sigma}(\phi)$ be the action chosen by player i when play is defined by ϕ. For example, if $\sigma = 1, 2, 3, \ldots P$ then $Out_1^{\sigma}(\phi) = \phi_1$ (which is a fixed strategy independent of the other players' moves), $Out_2^{\sigma}(\phi) = \phi_2(\phi_1)$, $Out_3^{\sigma}(\phi) = \phi_3(\phi_2(\phi_1), \phi_1)$, and so on. Given a vector of strategies ϕ, the payoff is $G(Out^{\sigma}(\phi))$.

For any r, $a_{\sigma^{-}(r)}$, and $\phi_{\sigma^{+}(r)} \in \mathcal{G}^{\sigma,r}$, consider the game played by the players in $\sigma(r)$.

[17] Note that we do not allow these response functions to be mixed strategies.

They see the strategies of the players in $\sigma^-(r)$ as fixed (at $a_{\sigma^-(r)}$), and see the strategies of the players in $\sigma^+(r)$ as a function of their joint action. Thus, to the players in $\sigma(r)$ the game has payoffs of the form:

$$G^{r,\sigma}(a_{\sigma(r)}; a_{\sigma^-(r)}, \phi_{\sigma^+(r)}) = G(a_{\sigma(r)}, a_{\sigma^-(r)}, \phi_{\sigma^+(r)}(a_{\sigma(r)}, a_{\sigma^-(r)})).$$

Given an order and any solution concept $\Upsilon(G, A)$ we now define the set $\Upsilon_{\sigma(r)}(G, A, \sigma)$ inductively. For all $a_{\sigma^-(r)} \in A_{\sigma^-(r)}$, let

$$\Upsilon_{\sigma(r)}(G, A, \sigma; a_{\sigma^-(r)}) = \bigcup_{\phi_{\sigma^+(r)} \in \Upsilon_{\sigma^+(r)}(G,A,\sigma)} \Upsilon(G^{r,\sigma}(a_{\sigma(r)}; a_{\sigma^-(r)}, \phi_{\sigma^+(r)}), A_{\sigma(r)})$$

where the union is over all response functions $\phi_{\sigma^+(r)}$ whose image $Out(\phi_{\sigma^+(r)})$ lies in the set $\Upsilon_{\sigma^+(r)}(G, A, \sigma)$. Let $\Upsilon_{\sigma(r)}(G, A, \sigma)$ be the set of $\phi_{\sigma(r)}$ such that $\phi_{\sigma(r)}(a_{\sigma^-(r)}) \in \Upsilon_{\sigma(r)}(G, A, \sigma; a_{\sigma^-(r)})$.

For a strategy set $B \subseteq \mathcal{G}^\sigma$ define the set of reachable actions by

$$R^\sigma(B) = \{a \in A \mid \exists \phi \in B \;\; s.t. \;\; a = Out^\sigma(\phi)\}$$

We propose that the set $R^\sigma(\Upsilon^\infty(G, A, \sigma))$ represents a possible solution concept for a game with ordering σ. We can now define the set of Stackelberg-Υ actions, denoted by $S^\Upsilon(G, A)$, of a game $< G, A >$.

Definition 11 *The set of Stackelberg-Υ actions $S^\Upsilon(G, A)$ of a game $< G, A >$ is given by:*

$$S^\Upsilon(G, A) = \bigcup_{\sigma \in \Sigma(\mathcal{P})} R^\sigma(\Upsilon^\infty(G, A, \sigma))$$

3.5.2 A Conjecture and a Question

A possible conjecture is that the correct solution concept for reasonable learners in a distributed setting is S^Υ where Υ is the "correct" solution concept for reasonable learners in a "synchronous game". If this is true, then the only impact of asynchrony is in separating timescales as in the proof of Theorem 6, while if it is false, it implies that the effect of asynchrony is more subtle.

First we note some relationships between the various solution concepts.

Lemma 1 *For any solution concept $\Upsilon(G, A)$ the following hold:*

i) $\Upsilon(G, A) \subseteq S^{\Upsilon}(G, A)$,

ii) $\Upsilon(G, A) \subseteq \tilde{\Upsilon}(G, A)$ *for all* $(G, A) \Rightarrow S^{\Upsilon}(G, A) \subseteq S^{\tilde{\Upsilon}}(G, A)$ *for all* (G, A),

iii) $S^{O^\infty}(G, A) = O^\infty(G, A)$.

Proof: i) This follows immediately since the order $\sigma \in \Sigma(\mathcal{P})$ with $\sigma(1) = \mathcal{P}$ shows that the Stackelberg version of Υ must contain Υ.
ii) This follows immediately from the definition of $S^{\Upsilon}(G, A)$.
iii) The relation $O^\infty(G, A) \subseteq S^{O^\infty}(G, A)$ follows from part (i), and we now show that the reverse holds. Assume $a_i \notin O^\infty(G, A)$; from the definition of O if a_i is overwhelmed by another action, then it must be overwhelmed for any subset of the other players actions. Therefore, $a_i \notin S_i^{O^\infty}(G, A)$ proving the equality. $\square$

The Stackelberg-Υ solution concepts are a way to take a "synchronous" solution concept Υ and generalize it to a setting with arbitrary asynchrony. Thus, we propose the Stackelberg-Υ solution concepts as a possible candidate for a decentralized solution concept $\mathcal{C}(G, A)$. The obvious question, then, is what synchronous solution concept Υ is appropriate. Foster and Vohra (1996) show that the appropriate solution concept for calibrated learners is the set of correlated equilibria. Let $Corr(G, A)$ represent the support of the set of correlated equilibria. If reasonable learners, rather than calibrated ones[18], also fill out the space of correlated equilibria then the following conjecture may be true.

[18]The standard form of calibrated learning algorithms are not responsive, so the question is whether the Foster and Vohra result holds for the responsive versions of such learning algorithms; such algorithms were simulated in Greenwald, Friedman, and Shenker (1998).

Conjecture 1 $S^{Corr}(G,A) \subseteq \mathcal{C}(G,A)$.

We call $S^{Corr}(G,A)$ the Stackelberg correlated set. In essence, this conjecture says that while we do not know what the correct solution concept is for "synchronous games," we suspect that it contains the set $Corr(G,A)$, and we further conjecture that the set $S^{Corr}(G,A)$ captures the effects of asynchrony. On the other hand, the set $D^{\infty}(G,A)$ is usually taken to be a superset of the actual asymptotic play in "synchronous" games. If that is indeed true, then it leads to the following question.

Question 1 *Is* $\mathcal{C}(G,A) \subseteq S^{D^{\infty}}(G,A)$*?*

We call $S^{D^{\infty}}(G,A)$ the Stackelberg undominated set. We have noted before that the possible disparity in learning rates leads to Stackelberg-like phenomena. If the only effect of asynchrony (added to our definition of reasonability) is to produce these Stackelberg-like phenomena, then this conjecture will be true and in fact we would have that $\mathcal{C}(G,A) = S^{\Upsilon}(G,A)$ for some solution concept Υ. We leave this as an open question which requires further investigation.[19]

3.6 Example

To get a more concrete sense of these solution concepts, recall the game discussed at the beginning of this section.

	L	R
T	1,1	.3,.6
B	.6,.3	0,0

Note that for the above game there are three orders $(\{1,2\})$,$(\{1\},\{2\})$, and $(\{2\},\{1\})$. For the order $(\{1,2\})$ we just have the original game which is dominance solvable with

[19]While our search for a tight solution concept has not yet succeeded, we are not alone. There are few solution concepts which have been proved to be tight for a class of learners. For example, various conditions have been shown to hold for fictitious play, but no tight solution concept is known. The only (nontrivial) example we know of is the tightness of correlated equilibria for calibrated learners (Foster and Vohra 1996).

actions (T, L), while for $\sigma = (\{1\}, \{2\})$ if player 1's action is fixed then player 2's only undominated strategy is $\phi_2(T) = L$ and $\phi_2(B) = L$, and after restricting to this, player 1's only undominated strategy is $s_1 = T$, thus the outcome for this game is (T, L), which is the same outcome for the order $\sigma = (\{2\}, \{1\})$, by symmetry. Thus, $S^{Corr}(G, A) = S^{D^\infty}(G, A) = (T, L)$ which is the same as $D^\infty(G, A)$, whereas O^∞ is the entire game.

4 Solvability and Mechanism Design

4.1 Solvable Games

Often the sets of play in the various solution concepts such as $S^{Corr}(G, A)$ or $O^\infty(G, A)$ are quite large, and in those cases one cannot predict with precision the asymptotic play of reasonable learners. There are, however, some games where the outcome is unambiguous. We will call such games solvable.

Definition 12 *A game $< G, A >$ is O-solvable if $|G(O^\infty(G, A))| = 1$. Similarly, a game $< G, A >$ is SC-solvable if $|G(S^{Corr}(G, A))| = 1$, it is SD-solvable if $|G(S^{D^\infty}(G, A))| = 1$, it is C-solvable if $|G(Corr(G, A))| = 1$, and it is D-solvable if $|G(D^\infty(G, A))| = 1$.*

Note that solvability does not require that there is a single eventual play, only that there is a single eventual outcome (payoff vector). Because $Corr(G, A) \subseteq S^{Corr}(G, A) \subseteq O^\infty(G, A)$ and $Corr(G, A) \subseteq D^\infty(G, A) \subseteq S^{D^\infty}(G, A) \subseteq O^\infty(G, A)$, any O-solvable game is both SC-solvable and SD-solvable, and any SD-solvable game is D-solvable and C-solvable. (See Lemma 1.)

Below is an example of a 3×3 game with varying degrees of solvability as x varies.

	L	C	R
T	4,6	5,4	1,1
M	1,5	6,4	5,2
B	2,2	3,5	3,x

When $x = 1$, this game is O-solvable, and when $x = 4$ it is SD-solvable (and SC-solvable) but not O-solvable. When $x = 7$, this game is not even D-solvable (or C-solvable).

To illustrate a more general O-solvable game, we define the class of *generalized serial* games $< G, A >$, following Moulin and Shenker (1992), to be those that have the following five properties for any i, j with $i \neq j$:

- Ordered action domains: $A_i \subseteq \Re$
- Cross-Monotonicity: $G_i(a) \geq G_i(\tilde{a}_j, a_{-j})$ for any $\tilde{a}_j \geq a_j$, $i \neq j$.
- Seriality: $G_i(a_j, a_{-j}) = G_i(\tilde{a}_j, a_{-j})$ for any $a_j, \tilde{a}_j \geq a_i$, $i \neq j$.
- Unique best reply: for each a_{-i} there exists an element $BR_i(a_{-i})$ such that

$$x_i \neq BR_i(a_{-i}) \Rightarrow G_i(BR_i(a_{-i}), a_{-i}) > G_i(x_i, a_{-i})$$

- Seriality of best reply: $BR_i(a_{-i}) = BR_i(\tilde{a}_j, a_{-ij})$ for any $\tilde{a}_j \geq BR_i(a_{-i})$

Theorem 10 *Generalized serial games are O-solvable.*

Proof: Since the O operator is monotonic, the iteration process must converge to a nontrivial fixed point. Let this fixed point of O be denoted by $I = (I_1, I_2, \ldots, I_n)$ with $\bot_i$ denoting the minimal element of I_i and $\top_i$ denoting the maximal element of I_i, and $\bot$ and $\top$ denoting the vectors of these extremal elements. Let $MAX_i(x_i) = max_{a_{-i} \in I_{-i}} G_i(x_i, a_{-i})$, and $MIN_i(x_i) = min_{a_{-i} \in I_{-i}} G_i(x_i, a_{-i})$. For any $a \in I$ and for any $x_i \in I_i$, $G_i(x_i, \top_{-i}) \leq G_i(x_i, a_{-i}) \leq G_i(x_i, \bot_{-i})$ so $MAX_i(x_i) = G_i(x_i, \bot_{-i})$ and $MIN_i(x) = G_i(x_i, \top_{-i})$. Assume that I is not a singleton, so the set $\{i | \bot_i < \top_i\}$ is nonempty. We can define i as the element in this set with the smallest $\bot_i$: $\bot_j < \top_j \Rightarrow \bot_j \geq \bot_i$. In particular, $G_i(\bot_i, \top_{-i}) = G_i(\bot)$, so $MIN_i(\bot_i) = MAX_i(\bot_i)$. If there exists some $x_i \in I_i - \bot_i$ such that $G_i(x_i, \bot_{-i}) < G_i(\bot)$, then $MAX_i(x_i) < MIN(\perp_i)$ and so $\perp_i$ overwhelms x_i. If there exists some $x_i \in I_i - \perp_i$ such that $G_i(x_i, \top_{-i}) > G_i(\perp_i, \top_{-i}) = G_i(\perp)$, then $MIN_i(x_i) > MAX(\perp_i)$ and so x_i overwhelms $\perp_i$. Thus, we must have $G_i(x_i, \top_{-i}) \leq G_i(\perp_i, \top_{-i}) = G_i(\perp)$ and $G_i(x_i, \perp_{-i}) \geq G_i(\perp)$ for all $x_i \in I_i - \perp_i$. Consequently, $BR_i(\top_{-i}) = \perp_i$ and $BR_i(\perp_{-i}) \neq \perp_i$. This contradicts the seriality of the function BR_i. $\square$

In Section 4.3 we will encounter examples of such generalized serial games. Another solvable game arises when rationing a fixed amount C of some good when all utilities are single-peaked (see, for example, Sprumont 1991). Let p_i be the location of agent i's peak. The uniform game can be defined as follows. Each agent announces a request a_i. If $\sum_i a_i \geq C$ then the allocations q_i are given by $q_i = \min[\lambda, a_i]$ where λ is the unique value such that $\sum_i q_i = C$. If $\sum_i a_i \leq C$ then the allocations q_i are given by $q_i = a_i + \lambda$ where λ is the unique value such that $\sum_i q_i = C$. In the case where $a_i = p_i$, the resulting allocation reduces to the uniform mechanism.

Theorem 11 *The uniform game is SD-solvable (and SC-solvable) but not O-solvable.*

Proof: First, we prove that the uniform game is D-solvable. Let $D^\infty = I_1 \times I_2 \ldots I_P$ with $I_i = [l_i, u_i]$ denote the result of iterated elimination of dominated actions. Note that $l_i \leq p_i \leq u_i$ since each agent gets the highest payoff by announcing p_i. Assume k is such that $l_i < l_k \Rightarrow l_i = u_i$. If $l_k \geq \frac{C}{P}$ then each action vector in D^∞ results in the same allocation with each agent getting $q_i = \frac{C}{P}$. Assume, to the contrary, that $l_k < \frac{C}{P}$. If $l_k < p_k$ then p_k dominates l_k (the allocations are monotonic in r_k and are strictly monotonic in the vicinity of l_k). If $l_k < \frac{C}{P}$ and $l_k = p_k$ then p_k dominates u_k (the allocations are monotonic in r_k and are strictly monotonic in the vicinity of u_k). Therefore, by contradiction, there can be no such k, and so all sets I_i are merely the singleton p_i.

Note that this proof show that if we held some of the actions fixed (not necessarily at their peak), then the D^∞ set of the game among the remaining players converges to the singleton (with each player's peak p_i the only remaining action). Since on each subgame the set D^∞ converges to the same singleton, the construction used in the Stackelberg undominated set also reduces to that singleton.

Next, we show that the uniform game is not O-solvable. Denote by $[l(r_i), u(r_i)]$ the set of player i's allocations (not payoffs) resulting from announcing action r_i, and letting the other actions vary from 0 to ∞. Both l and u are monotonically increasing in r_i, and $l(0) = 0$, $u(0) = \frac{C}{P}$, $l(\infty) = \frac{C}{P}$, $u(\infty) = C$. Because $u(0) = l(\infty)$, $[l(r_i), u(r_i)] \cap [l(r_i'), u(r_i')] \neq \emptyset$ for all r_i, r_i'. The allocation intervals always overlap, and so the payoff sets for any two actions overlap, so there are no overwhelmed actions: O^∞ is the entire strategy space for this game. $\square$

Our final example is that of *ordered externality games* (Friedman 1996 and 1997). These are nonatomic games where agents, labeled by a parameter ν, decide to participate (setting $a(\nu) = 1$) or not (setting $a(\nu) = 0$); if they participate, their payoff depends only on the size λ of the participating population (and if they don't participate, their payoff is zero). The payoffs decrease with the level of participation. Thus, for a given vector of actions, the payoffs are of the form $U_\nu(a(\nu), \lambda(a))$ which is nonincreasing in λ, and $U_\nu(0, \lambda(a)) = 0$. It is shown in Friedman (1995) that this game is O-solvable if and only if it converges under best-reply dynamics.

For example, consider the congestion game discussed in the Introduction played by a large number of players. Each player decides whether to send a packet of information. Let $\lambda(a)$ be the total number (measure) of players who decide to send a packet. The delay to a player is $D_\mu(\lambda)$ which is nondecreasing in λ, where μ is the capacity of the link. (For an M/M/1 FIFO queue, $D_\mu(\lambda) = \mu/(\mu - \lambda)$ for $\lambda < \mu$ and ∞ otherwise.) Thus the payoff to a player who sends a packet is $v - c(D_\mu(\lambda))$ where v is the personal value of the packet and $c(\cdot)$ is the delay cost, which is assumed to be nondecreasing. The payoff is 0 if the player does not send a packet. For many typical queuing processes, this game converges under best reply dynamics if the capacity of the queue, μ, is sufficiently large. (See Friedman and Landsberg 1993 for details.) Thus, in this case the game is O-solvable. These results also apply to similar congestion games with multiple links and players at different locations with λ becoming a vector depending on the type and location of player.

4.2 Implications for Mechanism Design on the Internet

So far, in our discussions of learning and convergence, we have implicitly assumed that the game is exogenously given. However, in the Internet, and in other distributed contexts, one would want to design the game in order to shape the nature of the resulting play and thereby achieve certain social goals. This is the mechanism design, or implementation, paradigm. To fix notation, consider an allocation problem with P agents. Let $\mathcal{U}$ denote the domain of utility functions (assumed, for the sake of simplicity, to be the same for each agent), and let $\mathcal{O}$ denote the set of possible outcomes. A social choice function is a mapping $F : \mathcal{U}^P \mapsto \mathcal{O}$. A mechanism is a set of action spaces A_i and a mapping[20] $M : A \mapsto \mathcal{O}$. Associated with each mechanism $< M, A >$ and a utility profile U is a (stable) game $G : A \mapsto \Re^P$ defined by $G_i(a) = U_i(M(a))$. We denote by $\mathcal{C}_M(U) \subseteq A$ the solution concept for a mechanism M at a particular utility profile U. A mechanism $< M, A >$ *implements* a social choice function F if $M(a) = F(U)$ for all $a \in \mathcal{C}_M(U)$.[21]

We now ask: which social choice functions can be implemented in a distributed setting? To be more precise: for which F's is there a mechanism $< M, A >$ such that $M(a) = F(U)$ for all $a \in \mathcal{C}_M(U)$? Since we do not know the exact nature of $\mathcal{C}$, we cannot answer this question definitively; however, we do have some partial results. Before presenting these

[20]Note that the set A is not necessarily the "natural" action space on the network, but is more commonly denoted the message space. For example, in the congestion game A could include a priority request, along with a transmission rate.

[21]This is sometimes called *strong* implementation in the literature.

results, we need a few definitions.

Definition 13 *Consider any pair $U, V \in \mathcal{U}^P$ and define $E = \{i | U_i \neq V_i\}$. F is* weakly coalitionally strategy-proof *(WCSP) if, when E is nonempty, there always exists some $j \in E$ such that $U_j(F(V)) \leq U_j(F(U))$. F is* strictly coalitionally strategy-proof *(SCSP) if $F(U) \neq F(V) \Rightarrow$ there exists $j \in E$ such that $U_j(F(V)) < U_j(F(U))$. F is* strictly strategy proof *(SSP) if $F(U) \neq F(V_i, U_{-i}) \Rightarrow U_i(F(U)) > U_i(F(V_i, U_{-i}))$. F is* Maskin monotonic *(MM) if $F(V) = F(U)$ whenever $U_i(x) \leq U_i(F(U)) \Rightarrow V_i(x) \leq V_i(F(U))$ for all allocations $x \in \mathcal{O}$ and all i.*

WCSP[22] merely requires that not all members of the deviating coalition can strictly gain by deviating. SCSP requires that there is no other outcome that is equivalent or better, in the eyes of the deviating coalition, to the truthful outcome. SSP requires that, for an individual deviator, no other outcome is equivalent or better. Thus, for an SSP social choice function F, the truth is a strict Nash equilibrium of F (though perhaps not the only Nash equilibrium), while for an SCSP social choice function F, the truth is a strict strong equilibrium (though, again, perhaps not the only one). Note that the definition of SSP implies nonbossiness (in fact, coalitional nonbossiness) when applied to a private goods context. Maskin (1985) proved that if F is Nash implementable (in the sense we mean here), then F is Maskin monotonic.

We do not yet have a tight definition of the solution concept $\mathcal{C}$, and so below we present results for implementation with different possible solution concepts. If a social choice function is implementable with a solution concept v we say it is v-implementable. We can now state our first theorem, that holds if the solution concept is indeed the upper bound O^∞.

Theorem 12 *If a social choice function F is O-implementable, then it must be SCSP.*

[22] This is also referred to as Group Strategy-Proof; see Muller-Satterthwaite (1985) .

Proof: Consider some mechanism $M : A \mapsto \mathcal{O}$ that implements F. Assume, to the contrary, that F is not SCSP. Then, there exists two utility profiles U and V such that $F(U) \neq F(V)$ but $U_i(F(U)) \leq U_i(F(V))$ for all i such that $U_i \neq V_i$. Let $E = \{i | U_i \neq V_i\}$. Since M implements F, there must be two action vectors u and v in A such that $M(u) = F(U)$ and $M(v) = F(V)$ and each are in the solution concepts at the respective utility profiles U and V; i.e., $u \in \mathcal{C}_M(U)$ and $v \in \mathcal{C}_M(V)$. Since $F(U) \neq F(V)$, we have $v \notin \mathcal{C}_M(U)$ and $u \notin \mathcal{C}_M(V)$. At the utility profile U, consider the Stackelberg ordering with elements in E leading: $\sigma = E, \mathcal{P} - E$. The allocations that result from this Stackelberg game must be the allocation $F(U)$, but the allocation $F(V)$ is different from $F(U)$ yet gives all the elements in E at least as good outcomes. Recall that $S^{O^\infty} = O^\infty$, so we can apply the solution concept O^∞ to the players in E, assuming that the agents in $\mathcal{P} - E$ are responding to these plays. The solution concept O^∞ applied to the game played by the agents in E contains the point u_E. Therefore, it must also contain v_E since the payoffs for v_E Pareto dominate the payoffs for u_E (and therefore none of the strategies in v_E are overwhelmed). Thus, the point v must be included in the solution set $\mathcal{C}_M(U)$, which contradicts our earlier result. $\square$

Note that the coalitional aspects of the O^∞ solution concepts, and hence the coalitional requirements of SCSP, did not arise because of some explicit notion of collusion among agents in our distributed setting. It arose because of the asynchrony where there could be multiple agents with long timescales, even though there was no explicit collusion.

Our next result is a slight extension of the original observation due to d'Aspremont and Gérard-Varet (1980) on Stackelberg-Solvable games.

Theorem 13 *If a social choice function F is $\mathcal{C}$-implementable, then F must be SSP.*

Proof: Assume, to the contrary, that there exists a social choice function F that is $\mathcal{C}$-implementable, with $< M, A >$ as the implementing mechanism, but for which there exists

U and V_i such that $F(U) \neq F(V_i, U_{-i})$ but $U_i(F(U)) \leq U_i(F(V_i, U_{-i}))$. Without loss of generality, assume $i = 1$ and consider a strict Stackelberg ordering with $\sigma = \{1, 2, 3, \ldots, P\}$. All points in the solution concept $\mathcal{C}_M(U)$ are mapped, by M, into $F(U)$; similarly, all points in $\mathcal{C}_M(V_1, U_{-1})$ are mapped, by M, into $F(V_1, U_{-1})$. Let u be some Stackelberg equilibrium with order σ in $\mathcal{C}_M(U)$, and let v be some Stackelberg equilibrium with order σ in $\mathcal{C}_M(V_1, U_{-1})$. Then, the payoff for agent 1 at v is at least as great as the payoff at u, and we can choose agent 1's learning algorithm to favor v_1 over u_1 (such as in the ρ-prioritized stage learners). Since v_{-1} is the Stackelberg response to v_1 and u_{-1} is the Stackelberg response to u_1 by construction, v must also be in the set $\mathcal{C}_M(U)$, as it is a possible outcome of the learning process. This contradicts our original assumption. $\square$

The following is a standard result about SSP and Maskin Monotonicity (for convenience we include the trivial proof):

Theorem 14 *If a social choice function F is SSP, then F must be Maskin Monotonic.*

Proof: Consider an SSP social choice function F and some V_i such that $U_i(x) \leq U_i(F(U)) \Rightarrow V_i(x) \leq V_i(F(U))$ for all allocations x. Assume, to the contrary, that $F(U) \neq F(V_i, U_{-i})$. Because F is SSP, we must have $V_i(F(V_i, U_{-i})) > V_i(F(U))$ and $U_i(F(V_i, U_{-i})) < U_i(F(U))$. This contradicts our assumption about V_i. $\square$

This leads immediately to the following Corollary:

Corollary 3 *If a social choice function F is $\mathcal{C}$-implementable, then F must be Maskin Monotonic.*

Note that in certain restricted domains, Maskin Monotonicity implies WCSP (see Shenker 1993 and Barbera and Jackson 1995 ; see Dasgupta, Hammond, and Maskin (1979) for a definition of a monotonically closed domain):

Theorem 15 *If a social choice function F is Maskin Monotonic, and the domain is monotonically closed, then F is WCSP.*

This leads to the following Corollary:

Corollary 4 *If F is $\mathcal{C}$-implementable and the domain is monotonically closed, then F is WCSP.*

Note that many of the most notable strategyproof mechanisms do not have any degree of resistance to coalitional manipulations. For instance, the Clarke-Groves (Clarke 1971, Groves 1973) mechanisms are not, in general, weakly coalitionally strategyproof.

4.3 Examples

We now discuss a few SD-implementable and O-implementable social choice functions and their implementing mechanisms.

The first example is the uniform social choice function; its S^{D^∞}-implementability follows trivially from Theorem 11. Since the uniform mechanism relies only on the peaks of the preferences, there is no real distinction between the uniform game and the uniform social choice function. Thus, Theorem 11 implies that the uniform social choice function is SD-implementable, because the direct mechanism is itself S^{D^∞}-solvable. While we have shown that the direct mechanism is not itself O-solvable, it remains on an open question as to whether the uniform social choice function is O-implementable through some other mechanism.

The second example comes from the congestion game with strictly monotonic (increasing in r_i, decreasing in c_i) and concave utilities $U_i(r_i, c_i)$ and a strictly convex constraint function f. The serial mechanism (see Moulin and Shenker 1992 for a description) can be described as follows.[23] When the agents are labeled so that $r_i \leq r_{i+1}$ for all i, the congestions c_i are recursively determined by the equation:

$$(\sum_i^{k-1} c_i) + (n-k+1)c_k = f(\sum_i \min[r_i, r_k])$$

We have the following theorem:

Theorem 16 *The serial mechanism, with strictly monotonic and concave utilities and a strictly convex constraint function f, is a generalized serial game.*

Proof: Consider some i and some $j \neq i$. The payoff $G_i(r) = U_i(r_i, c_i(r))$ is monotonic in r_j since $c_i(r)$ is monotonic in r_j and U_i is monotonic in c_i. Moreover, from the construction it is clear that $c_i(r) = c_i(r_{-j}, \hat{r}_j)$ for all $r_j, \hat{r}_j \geq r_i$, so the same holds for the payoffs $G_i(r)$. Consider the function $g(x) = G_i(r_{-i}, x) = U_i(x, c_i(r_{-i}, x))$. Since U_i is convex, and the opportunity set $(x, c_i(r_{-i}, x))$ is strictly concave, there is a unique point of tangency, and so the game has unique best replies $BR_i(r_{-i})$. Lastly, consider some agent j such that $r_j \geq BR_i(r_{-i})$. Varying r_j changes the opportunity set $(x, c_i(r_{-i}, x))$, but the tangent at $x = BR_i(r_{-i})$ remains unchanged. Therefore, the best reply remains unchanged. □

Therefore, the serial mechanism is O-solvable in this setting. Define the serial social choice function as the allocation resulting from the (unique) Nash equilibrium of this game. This social choice function is obviously O-implementable.

Corollary 5 *The serial mechanism, with strictly monotonic and concave utilities and a strictly convex constraint function f, is O-solvable.*

4.4 Discussion

One might ask why, if one can only implement strategyproof social choice functions, does one bother with the mechanism design paradigm at all. Why not always use the direct method –

[23]The serial mechanism is a formalization of the fair queuing packet scheduling algorithm in routers (Demers et. al. 1990); variants of fair queuing are currently implemented on some Internet routers.

asking for utilities to be revealed and then applying F – instead of using an indirect mechanism M. In the former case you can utilize the focal point nature of truthful revelation, and can implement all strategyproof social choice functions, whereas in the indirect method one can only implement a narrower class of social choice functions (SSP and Maskin Monotonic).

While in many cases it is obviously preferable to use direct methods, there are occasions where indirect mechanisms are preferable. In some contexts the utility functions are very complex, and revealing them involves significant communication overhead. For instance, the performance of a video application is not a simple function of, say, the average and variance of the packet delays; instead, the performance depends on the exact string of packet delays.[24] In such cases, the ability to use indirect mechanisms with their substantially less complex signaling is a significant advantage.

In addition, and perhaps more fundamentally, in many network situations the agents do not know their exact utility functions. Agents can compare two different levels of service and decide with which they are happier, but they cannot abstractly represent these trade-offs without actively experiencing them. For instance, the optimal trade-off between bandwidth and delay in a video stream for an agent will depend on many details of the particular instance – such as the particular scene being transmitted, the exact delay distribution, and the clarity of speech – and quantifying this relationship beforehand is quite impractical. To use an analogy, specifying the exact utility function of such network applications is much like trying to specify the optimal contrast setting on a television set. Since the optimal contrast setting depends on many details, such as the lighting in the room and the darkness of the scene, most users could not accurately articulate the underlying utility function; most of us merely turn the contrast knob until we notice that any deviation from that setting produces worse results. Similarly, in many networking situations, users can compare their

[24]Video applications adapt their playback point in response to the observed delays, and the performance of the application depends on the behavior of this playback point. See Bajaj et al. (1998)

satisfaction at two different levels of service that they have actually experienced, but they typically cannot provide a formal expression of their utility function. Users should be given an ability to adjust parameters (controlling bandwidth-delay trade-offs, or resolution-loss trade-offs, etc.) instead of having to specify a utility function directly. That means that we are forced to use indirect methods whereby users both 'learn' the equilibrium but also learn about their own preferences.

5 Related Work

The notion of learning through repeated play has a long history, early papers include Robinson (1951) and Brown (1951), and perhaps even Cournot (1838). The subsequent literature is vast and far beyond our capability to summarize in this section. We refer interested readers to the set of notes by Fudenberg and Levine (1996) for a comprehensive survey. As we discussed in Section 1, much of the recent work has focused on such learning algorithms that predict the actions of opponents, and then myopically optimize – perhaps approximately as in stochastic fictitious play – with respect to those predictions. These prediction methods investigated can take on a particular form (e.g., Bayesian, as in Kalai and Lehrer 1993, calibration as in Foster and Vohra 1996 or consistency, as in Fudenberg and Levine 1995). The overwhelming forecast of this line of research is that such learners end up playing in either correlated equilibria or Nash equilibria. This branch of the literature is appropriate to situations where players can observe the actions of others, and know the payoff function, and thus do not apply to the more distributed situations we are concerned with here.

There is another branch of the literature that deals with "low-rationality" approaches to learning. Relevant work along this line includes Roth and Erev (1995), Erev and Roth (1996), Borgers and Sarin (1995, 1996), Mookerji and Sopher (1994), Van Huyck et al. (1996) to name a few. The information assumed to be available to the players is roughly consistent with what we assume for our distributed systems; users are not given any information beyond the payoffs they receive. The *reinforcement*, or *stimulus-response*, learning algorithms discussed in this literature are quite similar in spirit to the examples we give here. The main technical difference is that we impose the requirement of responsiveness, which usually does not arise in this literature, although the algorithm of Roth and Erev (1996), for certain parameter values, is an exception. However, a more important distinction is that much of this literature is focused on comparing the behavior of particular learning algorithms to experimental results. We make no claim for a special role for any particular member of the class of reasonable learners. On the Internet, learning algorithms, rather than being the product of inherent mental processes which may have some universal properties, are typically manually constructed and embedded in programs and thus can change over time and differ between machines. As a result, in this paper we instead have identified a basic *reasonableness* criterion that would apply to all such learning algorithms, and focus on the resulting solution concept.

The nonstandard nature of these solution concepts is due to the combination of responsiveness and asynchrony, by which we mean that the timescales on which different agents adjust their actions can vary widely. While the responsiveness requirement is somewhat foreign to the learning literature, it has long be known that various forms of asynchrony – such as the existence of "patient players" (see Fudenberg and Levine 1989), the ability to make commitments (see Rosenthal 1991), and the capacity to establish reputations (Watson 1993) – can all disrupt more traditional forms of equilibria. In the previous analyses, these "patient" players, or leaders, were seen as manipulating the system. Here, the asynchrony arises quite naturally out of the different time scales players have. There has been some analysis of games that are resilient to this form of manipulation. In particular, d'Aspremont and Gérard-Varet (1980) introduced the notion of "Stackelberg-Solvable" to refer to equilibria that were robust against the one agent committing to an action; they showed that only strategyproof social choice functions could be implemented with Stackelberg-Solvable equilibria. However, such work has not focused on the general solution concept that incorporates arbitrary forms of such asynchrony.

Lastly, we comment that there is empirical evidence supporting our theoretical analyses. The first is work by Chen (1997) who compared the learning behavior of human subjects playing a cost sharing game using both average cost sharing, a D-solvable game, and serial cost sharing, an O-solvable game. The games were played asynchronously and with limited information, essentially replicating the network setting we consider here except for the restriction to two players. Her work shows rapid and robust convergence to the unique Nash equilibrium in the O-solvable game, regardless of the degree of asynchrony, while the D-solvable game was strongly affected by asynchrony and showed less robust convergence. The second line of empirical work, described in Greenberg, Friedman and Shenker (1998), involves numerical simulations of 6 different reasonable learners chosen from the literature interacting in a network setting. This results are consistent with the theoretical analysis in this paper, in that play is observed outside of the traditional synchronous solution concepts and that in some games play is restricted to a strict subset of O^∞ (in the simple games considered, $S^{D^\infty} = S^{Corr}$ and so the simulations did not provide any indication of which, if either, of these solution concepts apply). Moreover, in these simulations, three factors were required to leave the synchronous solution concept: responsive learning algorithms, low information (players not knowing the payoff structure) and highly asynchronous play. When any one of these factors were removed, play was observed to converge to D^∞. Of course, these simulations in no way constitute proof of a general result; we mention them here to merely to give some intuition.

6 Open Questions

The design of the modern Internet is clearly an extremely important problem which will have important economic ramifications in many arenas. It is becoming increasingly clear that incentive properties are an important aspect of network design. In order to achieve socially desirable allocations of the Internet's resources, we will need to know the appropriate solution concept, and that will require an understanding of learning, and of the joint convergence of learning algorithms, in such distributed systems. This has been our focus in this paper. We freely admit that our treatment is far from complete, and many important questions still remain open.

The most obvious, and most compelling, open question centers on the nature of the tight solution concept for this class of learning algorithms. In particular, is Conjecture 1 true and what is the answer to Question 1? A closely related open question is how to tightly characterize the set of social choice functions that are implementable with this solution concept. While Theorem 13 and Corollary 3 gives necessary conditions, we suspect they may not be sufficient.

However, the necessary conditions Theorem 13 and Corollary 3 already severely limit the class of implementable social choice functions. While it is of academic interest to precisely describe the class in question, it is of more practical importance to broaden this class. Perhaps the ideas of *virtual implementation*, following the ideas of Abreu and Matsushima (1992), could allow us to virtually implement a much wider class of social choice functions with this solution concept. A complementary approach would be to attempt to design the Internet in a way that mitigates the learning difficulties, perhaps by supplying learners with more information about the structure of the game and the play of other players. Future work will determine whether these ideas are practical.

References

[1] D. Abreu and H. Matsushima. Virtual implementation in iteratively undominated strategies: complete information. *Econometrica*, 60:993–1008, 1992.

[2] W. B. Arthur. Designing economic agents that act like human agents: A behavioral approach to bounded rationality. *Learning and Adaptive Economic Behavior*, 81(2):353–9, 1991.

[3] S. Bajaj, L. Breslau, and S. Shenker. Is service priority useful in networks? In *Proc. ACM Sigmetrics '98*, 1998.

[4] S. Bajaj, L. Breslau, and S. Shenker. Uniform versus priority dropping for layered video. In *Proc. ACM Sigcomm '98*, 1998.

[5] S. Barbera and M. Jackson. Strategy-proof exchange. *Econometrica*, 63:51–88, 1995.

[6] A. Blumer, A. Ehrenfeucht, D. Haussler, and M. Warmuth. Learnability and the Vapnik-Chervonenkis dimension. *Journal of the Association for Computing Machinery*, 36(4), 1989.

[7] T. Borgers and R. Sarin. Learning through reinforcement and replicator dynamics. *Journal of Economic Theory*, 77(1):1–14, 1997.

[8] G. Brown. Iterative solutions of games by fictitious play. In T. Koopmans, editor, *Activity Analysis of Production and Allocation*. Wiley, New York, 1951.

[9] Y. Chen. Asynchronicity and learning in cost sharing mechanisms. Mimeo, 1996.

[10] M.S. Chrystall and P. Mars. Adaptive routing in computer communication networks using learning automata. In *Proc. IEEE Nat. Telecomm. Conf.*, pages 121–8, 1981.

[11] D. Clark, S. Shenker, and L. Zhang. Supporting real-time applications in an integrated services packet network: architecture and mechanism. In *Proceedings of Sigcomm 92*, pages 14–26, Baltimore, Maryland, August 1992. ACM. *Computer Communication Review*, Volume 22, Number 4.

[12] E. H. Clarke. Multipart pricing of public goods. *Public Choice*, 11:17–33, 1971.

[13] R. Cocchi, S. Shenker, D. Estrin, and L. Zhang. Pricing in computer networks: Motivation, formulation, and example. *Transactions on Networking*, 1(6), December 1993.

[14] A. Cournot. *Recherches sur les Principes Mathematics de la Theorie de la Richesse.* Hachette, Paris, 1838.

[15] P. Dasgupta, P. Hammond, and E. Maskin. The implementation of social choice rules. *Review of Economic Studies*, 46:153–170, 1979.

[16] C. D'Aspremont and L. Gérard-Varet. Stackelberg-solvable games and preplay communication. *Journal of Economic Theory*, 23:201–217, 1980.

[17] Alan Demers, Srinivasan Keshav, and Scott Shenker. Analysis and simulation of a fair queueing algorithm. *Journal of Internetworking*, 1(1):3–26, January 1990.

[18] J. Duggan and M. Le Breton. Dominance-based solutions for strategic form games. mimeo, 1997.

[19] I. Erev and A. Roth. On the need for low rationality cognitive game theory: reinforcement learning in experimental games with unique mixed strategy equilibria. Mimeo, 1996.

[20] D. Ferguson. *The Application of Microeconomics to the design of resource allocation and control algorithms in Distributed Systems.* PhD thesis, Dept. of Computer Science, Columbia University, New York, New York, 1989.

[21] D. Ferguson, C. Nikolaou, and Y. Yemini. An economy for flow control in computer networks. In *Infocom*, pages 110–118, Ottawa, Canada, April 1989. IEEE.

[22] D. Foster and R. Vohra. Calibrated learning and correlated equilibrium. Mimeo, 1996.

[23] D. Foster and R. Vohra. Regret in the on-line decision problem. Mimeo, 1997.

[24] E. J. Friedman. Dynamics and rationality in ordered externality games. *Games and Economic Behavior*, 16:65–76, 1996.

[25] E. J. Friedman. Learnability in non-atomic externality games, with applications to computer networks. mimeo., 1997.

[26] E. J. Friedman and A. S. Landsberg. Short run dynamics of multi-class queues. *Operations Research Letters*, 14:221–229, 1993.

[27] E. J. Friedman and S. Shenker. Synchronous and asynchronous learning by responsive learning automata. mimeo., 1995.

[28] D. Fudenberg and D. Levine. Reputation and equilibrium selection in games with a patient player. *Econometrica*, 57:759–778, 1989.

[29] D. Fudenberg and D. Levine. Steady state learning and Nash equilibrium. *Econometrica*, 61(3):547–573, 1993.

[30] D. Fudenberg and D. Levine. Conditional universal consistency. Mimeo, 1995.

[31] D. Fudenberg and D. Levine. Consistency and cautious fictitious play. *Journal of Economic Dynamics and Control*, 19:1065–1089, 1995.

[32] D. Fudenberg and D. Levine. Theory of learning in games. Mimeo, 1996.

[33] A. Greenwald, E. Friedman, and S. Shenker. Learning in network contexts: Experimental results from simulations. mimeo, 1998.

[34] T. Groves. Incentives in teams. *Econometrica*, 41(4):617–631, July 1973.

[35] A. Gupta, D. O. Stahl, and A. B. Whinston. Managing the Internet as an economic system. Technical report, Department of Economics, University of Austin at Texas, July 1994.

[36] W. Hoeffding. Probability inequalities for sums of bounded random variables. In N. Fisher and P. Sen, editors, *The Collected Works of Wassily Hoeffding*. Springer-Verlag, 1994.

[37] M. T. Hsiao and A. Lazar. A game theoretic aproach to decentralized flow control of markovian queueing networks. In Courtois and Latouche, editors, *Performance'87*, pages 55–73. North-Holland, 1988.

[38] J. Van Huyck, R. Battalio, and F. Rankin. Selection dynamics and adaptive behavior without much information. Mimeo, 1996.

[39] E. Kalai and E. Lehrer. Rational learning leads to Nash equilibria. *Econometrica*, 61(5):1019–1045, 1993.

[40] E. Kalai and E. Lehrer. Subjective games and equilibria. *Games and Economic Behavior*, 8:123–163, 1995.

[41] Y. A. Korilis and A. A. Lazar. On the existence of equilibria in noncooperative optimal flow control. CTR Technical Report 340-93-13, Center for Telecommunications Research, Columbia University, New York, April 1993.

[42] Y. A. Korilis, A. A. Lazar, and A. Orda. The designer's perspective to noncooperative networks. In *Infocom*, Boston, Massachusetts, April 1995.

[43] Y. A. Korilis, A. A. Lazar, and A. Orda. The role of the manager in a noncooperative network. In *Infocom*, San Fransisco, California, March 1996.

[44] R. Lagunoff and A. Matsui. An "anti-folk theorem" for a class of asynchronously repeated coodination games. Mimeo, 1995.

[45] J. Mackie-Mason and H. Varian. Pricing the internet. In B. Kahin and J. Keller, editors, *Public Access to the Internet.* ACM, Boston, Massachusetts, May 1993. version of February, 1994.

[46] J. Mackie-Mason and H.. Varian. Pricing congestible network resources. Technical report, University of Michigan, Michigan, USA, July 1994.

[47] E. Maskin. The theory of implementation in Nash equilibrium: A survey. In L. Hurwicz, D. Schmeidler, and H. Sonnenschein, editors, *Social Goals and Social Organization: Essays in Memory of Elisha Pazner*, chapter 6, pages 173–204. Cambridge University Press, Cambridge and New York, 1985.

[48] L.D. Mason and X.D. Gu. Learning automata models for adaptive flow control in packet-switching networks. In K.S. Narendra, editor, *Adaptive and Learning Systems*, pages 213–28. Plennum Press, New York, 1986.

[49] S. McCanne, V. Jacobson, and M. Vetterli. Receiver-driven layered multicast. In *Proceedings of Sigcomm 96*, Palo Alto, California, August 1996.

[50] H. Mendelson and S. Whang. Optimal incentive-compatible priority pricing for the m/m/1 queue. *Operations Research*, 38(5):870–883, September–October 1990.

[51] P. Milgrom and J. Roberts. Rationalizability, learning and equilibrium in games with strategic complementarities. *Econometrica*, 58:1255–1278, 1990.

[52] D. Mookherjee and B. Sopher. Learning and decision costs in experimental constant sum games. *Games and Economic Behavior*, 19(1):97–132, 1996.

[53] H. Moulin and S. Shenker. Serial cost sharing. *Econometrica*, 60:1009–1037, 1992.

[54] E. Muller and M. Satterthwaite. Strategy-proofness: the existence of dominant strategy mechansims. In L. Hurwicz, D. Schmeidler, and H. Sonnenschein, editors, *Social Goals and Social Organization: Essays in Memory of Elisha Pazner*, pages 131–171. Cambridge University Press, Cambridge and New York, 1985.

[55] John Murphy and Liam Murphy. Bandwidth allocation by pricing in ATM networks. Technical report, Dublin City University, Ireland, July 1994.

[56] J. Nagle. On packet switches with infinite storage. RFC 970, Internet Engineering Task Force, December 1985.

[57] Y. Rekhter, P., and S. Bellovin. Financial incentives for route aggregation and efficient address utilization in the internet. preprint, ATT, 1996.

[58] J. Robinson. An iterative method of solving a game. *Annals of Mathematics*, 54:296–301, 1951.

[59] R. Rosenthal. A note on the robustness of equilibria with respect to commitment opportunities. *Games and Economic Behavior*, 3:237–243, 1991.

[60] A. Roth and I. Erev. Learning in extensive form games: experimental data and simple dynamic models in the intermediate term. *Games and Economic Behavior*, 8:164–212, 1995.

[61] B. A. Sanders. An incentive compatible flow control algorithm for rate allocationin computer networks. *IEEE Transactions on Computers*, C-37(9):1067–1072, 1988.

[62] S. Shenker. Efficient network allocations with selfish users. In P. J. B. King, I. Mitrani, and R. J. Pooley, editors, *Performance '90*, pages 279–285. North-Holland, New York, 1990.

[63] S. Shenker. Some technical results on continuity, strategy-proofness, and related strategic concepts. preprint, 1993.

[64] S. Shenker. Making greed work in networks: A game-theoretic analysis of switch service disciplines. *IEEE/ACM Transactions on Networking*, 3:819–831, 1995.

[65] P.R. Shrikantakumar. A simple learning scheme for priority assignment in a single-server queue. *IEEE Trans. on Syst., Man, and Cybern.*, SMC-16:751–54, 1986.

[66] Y. Sprumont. The division problem with single-peaked preferences: a characterization of the uniform allocation rule. *Econometrica*, 59:509–520, 1991.

[67] B. Stinchcombe. Maximal strategy sets for continuous-time game theory. *Journal of Economic Theory*, 56(2):235–65, 1992.

[68] A. Tanenbaum. *Computer Networks*. Prentice Hall, Upper Saddle River, New Jersey, 1996.

[69] E.L. Thorndike. Animal intelligence: an experimental study of the associative processes in animals. *Psychol. Monogr.*, 2, 1898.

[70] L. Valiant. A theory of the learnable. In *Proceedings of the Sixteenth Annual ACM Symposium on Theory of Computing*, Washington, D.C., 1984.

[71] J. Watson. A 'reputation' refinement without equilibrium. *Econometrica*, 61:199–206, 1993.