When to Arrive in a Congested System: Achieving Equilibrium via Learning Algorithm

Parth Thaker, Aditya Gopalan and Rahul Vaze

***Abstract*—Motivated by applications in competitive WiFi sensing, and competition to grab user attention in social networks, the problem of when to arrive at/sample a shared resource/server platform with multiple players is considered. Server activity is intermittent, with the server switching between ON and OFF periods alternatively. Each player spends a certain cost to sample the server state, and the per-player payoff is inversely proportional to the number of simultaneously connected/arrived players. The objective of each player is to arrive/sample the server as soon as any ON period begins while incurring minimal sensing cost and to avoid having many other players overlap in time with itself. For this competition model, we propose a distributed randomized learning algorithm (strategy to sample the server) for each player, which is shown to converge to a unique non-trivial fixed point. The fixed point is moreover shown to be a Nash equilibrium of a game, where each player's utility function is demonstrated to possess all the required selfish tradeoffs.**

I. Introduction

A canonical example of a general problem "when to join a shared server queue to maximize a payoff" [1]–[3] is a concert queue problem [4], where a server (ticket window) opens and closes at fixed times, and the problem for each customer is to decide when to arrive at the server queue amidst many competing customers so as to minimize its sum of waiting and service times. In modern paradigms, as discussed next, the server opening and closing time need not be deterministic, and moreover, servers need not necessarily serve customers one at a time, and multiple customers can be simultaneously served but with a reduced individual perceived payoff or quality of service (QoS) that is inversely proportional to the number of concurrently existing customers.

Some of the applications that present with these new attributes are as follows. Consider a large wireless network with multiple access points (APs) deployed at fixed locations in a given geographical area. Mobile nodes move within the given area and encounter intermittent connectivity to APs depending on their locations. To discover whether any AP is within transmission range, a mobile node employs sensing, which is equivalent to deciding to arrive in the system, and which comes at a cost (battery usage). The mobile node would like to sense (arrive) when the least number of other users are connected to the same AP, since some form of fair sharing is typically employed by the AP. For example, a video might be available in SD with congestion, compared to HD otherwise.

A more modern application is on grabbing user attention in social media under competition [5], [6]. In a social network platform, user attention intensity varies over time, and multiple players (advertisers, users) compete to get as much utility (eyeballs, impressions) as possible. Treating user attention as a limited, time-varying, shared resource, each player has to decide when to tweet or insert their ads given that there are multiple such competing players, so that their tweets have the highest impact in the limited, high intensity user attention span, where each tweet incurs a certain cost. Players would want to avoid tweeting together with other players, since if multiple tweets are shared within a short span of time, each gets a divided impact/attention.

Another related paradigm is strategic job submission times for users to cloud services, where the server is always ON, but its price fluctuates depending on the demand process. Thus, each player wants to avoid submitting jobs together with other players (to incur lower price), subject to its deadlines.

To study the mentioned modern paradigms, we consider a single game model, where a shared server has intermittent activity periods, and transitions between active and inactive states in a stochastic manner. Following some strategy, each user senses the server and connects to it as soon as it discovers it to be in active state. Each sensing comes at a fixed cost, precluding sensing at arbitrarily small intervals. Once the user connects to the server, it stays connected till the time its service is completed or the server transitions to the inactive state, whichever is earlier. To model the congestion aspect, the perceived payoff or QoS for each user is inversely proportional to the number of other users encountered by it during the time it was connected/served to/by the server. Thus, to maximize its utility, each user has to decide on a strategy to sense (and connect to) the server as soon as it turns active, and also to encounter a minimal number of other users during its connectivity time. In prior work, Jeong et al. [7] and Kumar et al. [8] considered only one user without any competition, and found the exact optimal sensing distribution given the server activity distribution.

A. Related Work

In the traditional sequential service setting, explicit Nash equilibrium (NE) arrival distributions have been found for the concert queue problem by Jain et al. [4]. Another related model is that of distributed access in wireless (WiFi) networks, where multiple nodes contend for slots when they have packets to transmit, and transmissions are successful only if one node ends up transmitting. Tang et al. study the equilibrium question [9] for this setting when there is an exponential back-off mechanism for contention resolution.

Compared to prior work, the problem considered here covers a more general setting where a) there is uncertainty over server activity periods, b) repeated sensing/arrival (instead

978-3-901882-90-6/17 ©2017 IFIP

of single shot decision as in [4]), and c) multiple nodes are served at the same time from a shared resource.

Under strategic behavior from all competing players, without cooperation, a natural goal from a system design standpoint is to find stable operating points or equilibria from which no player would prefer to deviate. Accordingly, the objective of this paper is find an equilibrium strategy for each competing player in this model, where the strategy for a player is comprised of the decisions to sample the system or not, at instants of time. Towards this end, the typical approach is to identify a per-node utility function, and try to find a NE for it, if it exists. This approach is analytically intractable for the problem at hand for most choices of natural utility functions.

To make analytical progress, we take an alternate route of considering that the players are running a natural distributed learning algorithm that adjusts its sensing behavior dynamically in response to its perceived payoff thus far, and show that it reaches equilibrium.

Finding learning algorithms that achieve equilibrium has been considered for *congestion* games (that are also potential games) where the congestion costs are additive, and the multiplicative weights learning algorithms is known to converge to NE [10], [11]. For a more general setting, Friedman and Shenker [12] showed that learning algorithms can achieve the NE in a two player zero-sum game, however, a similar result does not hold for a three player game as shown by Daskalakis et al. [13]. For a brief survey, we refer the reader to the work of Shoham et al. [14]. For non-congestion games, learning algorithms achieving the NE has been briefly considered [15]–[17]. Learning algorithms have also been used to achieve NE in spectrum access games [16], [17]. The game considered in this paper is not a congestion game, and is inherently *repeated*, where each player has to make its decisions repeatedly.

B. Our Results

It is easy to argue that a deterministic sensing/arrival strategy cannot be an equilibrium solution for the considered game. Therefore, we consider that each player employs a randomized strategy for sensing, i.e., in each slot it senses with a certain probability. The learning algorithm we propose (to update the sensing probability) learns the platform/server activity period frequency by computing how often the server was found active in previous sensing attempts, and implements a form of congestion control by exponentially decreasing its sensing probability with the number of other competing nodes encountered by it. Thus, given the per-sensing cost, the algorithm adapts to strike a balance between missing out on server activity periods and encountering large number of other nodes, when it senses the server. The learning algorithm does not require explicit information about the other players' strategies or the total number of them, and only depends on its accumulated reward. It thus has a low 'learning overhead' compared to best response strategies.

The main result of this paper is to show that the proposed learning algorithm converges to a unique, non-trivial fixed point. We also explicitly characterize the fixed point, and show that it is in fact a NE for a sensing game in which each player's utility function has a particular form. As one would expect, the corresponding per-player utility is increasing in its own sensing probability, and decreasing in the other nodes' sensing probabilities and sensing costs.

To prove our results, we first consider an expected version of the learning algorithm, where all random variables are replaced by their expected values. We then find the underlying utility function that the expected learning algorithm is trying to maximize. Corresponding to this utility function, we identify a multiplayer game $\mathcal{G}$, and show that there is a unique NE for this game, and that is achieved by the best response strategy. To show the convergence of the actual learning algorithm to a fixed point, we show that its updates converge to the best response actions for $\mathcal{G}$. Some of the proof techniques used in this paper are similar to those of Tang et al. [9]; however, the specific proofs themselves are entirely different.

II. System Model and Preliminaries

Consider a time-slotted system, where a server alternates between two states $\{\text{ON, OFF}\}$ following a two-state Markov chain. The durations of each ON and OFF periods are assumed to be independent and geometrically distributed with parameters λ_c and λ_d, respectively. We partition the total slots into frames, where each *frame* consists of M consecutive time slots. Whenever convenient, we will use $k \geq 1$ to index frames, and $t \in \{1, 2, \dots, M\}$ to index slots within a frame; the double-index notation (t, k) will thus denote the t-th time slot in frame k.

Consider N players in a system, where player ℓ employs a probabilistic sensing strategy $\{p_\ell(k) : k \geq 1\}$, where $p_\ell(k)$ is the probability with which player ℓ senses the server in each slot within frame k, to check whether the server is in ON state. Each player incurs a cost c_s upon a sensing attempt.

If, on sensing, a player finds the server to be in the OFF state, then the player senses with the same probability in each slot until the end of that frame, and then updates the sensing probability in the next frame using (1). Alternatively, if the server is found to be in ON state, the player joins/connects to the server. The service time (number of slots needed for completion of service) for each player is assumed to be geometrically distributed with parameter μ. The player stays connected to the server until its service is completed, or till the time the server remains in the ON state, whichever is earlier. The case $\mu = \infty$ corresponds to player requiring unlimited connection. For $\mu < \infty$, a player's service is defined to be *successful* if its service is completed before the end of the ON period during which the player connected.

With this strategy, during an ON period, multiple players may discover the server to be in the ON state and connect to it, creating congestion for each other. The competition or congestion aspect is modelled by assuming that the each player's perceived payoff or QoS is inversely proportional to the number of connected players. Such a model is relevant for video broadcast, or Skype, etc. where quality suffers with congestion without necessarily changing the service times, e.g.

a video becomes available in SD with congestion, compared to HD otherwise. An alternate model of congestion is to change the parameter μ depending on the number of other active users, following a processor sharing schedule, however, that makes the problem too challenging. Thus, the inherent objective for each player is to maximize the number of successful service completions in time $[0, t]$ as $t \to \infty$ given the per-sensing cost of c_s while encountering as few other players as possible during its service times.

A typical approach to studying this game would be write a utility function for player ℓ, as $U_\ell = f(p_{k,k\in[1:N]}, N, c_s, \lambda_c, \lambda_d)$, where p_k is the strategy of player k, f is a decreasing function of N, $p_k, k \neq \ell$ c_s, and λ_d, and an increasing function of λ_c, and find its NE. Such an approach is fairly complicated and analytically intractable for this problem. Instead, we consider learning type algorithms to define the adaptive sensing strategies that can be shown to converge to a fixed point/equilibrium.

For a special case of $\mu = \infty$, one can directly identify the NE.

Lemma 1: With $\mu = \infty$, when each player stays connected to the server (as soon as it senses it to be ON) until it goes to the OFF state, the optimal strategy derived in [8] for only one user without any competition is a NE.

Proof: If all players other than ℓ are following the optimal strategy [8] for only one user when there is no competition, then its optimal for player ℓ to follow that strategy as well, since if it deviates from that either it reduces its own chances of connecting to the server or pays more sensing cost, while encountering the same number of other players. □

A. Learning/adaptive sensing strategy

Let the set of players be denoted by $\Gamma = \{1, 2, \dots, N\}$; we use the notation $\Gamma_{-\ell} = \Gamma \backslash \{\ell\}$ to denote the set of all players except player ℓ. Let $\mathbf{p}(k) \equiv (p_\ell(k))_{1\le\ell\le N}$ be the sensing vector employed by the N players. We assume the frame size M to be large, so that under a two-time scale decomposition, the sensing probability is updated slowly enough (i.e., before each frame starts), while at the same time allowing players to learn about, and adapt to, the underlying server ON-OFF process and other players' strategies.

Let the server be in the ON state at time slot t, where the server last transitioned into the ON state at time slot $t_c \le t$, i.e., the server is ON throughout $[t_c, t]$. A player is defined to be *active* at time slot t if it discovered the ON state in time period $[t_c,\ t]$ and its service is not finished by time slot t. We denote by $X(t) \in \{0, \dots, N\}$ the number of players that are active at time slot t. Let us tag a player $\ell \in \Gamma$ for the remainder of the discussion. For a fixed frame k, for the tagged player ℓ, let $\mathbf{1}_{\text{Sense}}(t)$ denote the indicator random variable that it senses at (t, k), $\mathbf{1}_{\text{S}}(t)$ the indicator random variable that the server is in ON state at time slot t in frame k, and $\mathbf{1}_\ell(t)$ the indicator random variable that the tagged player ℓ is active at time slot t. For the tagged player ℓ, at the end of the k^{th} frame, i.e., at slot (M, k), define the random variable $\hat{\mathcal{A}}(k)$ to be the empirical average of the number of players that were active (including itself) for any slot in frame k in which player ℓ was active in the system. Formally,

$$\hat{\mathcal{A}}(k) = \begin{cases} \frac{\sum_{t=1}^M \mathbf{1}_\ell(t) X(t)}{\sum_{t=1}^M \mathbf{1}_\ell(t)} & \text{if } \sum_{t=1}^M \mathbf{1}_\ell(t) > 0, \\ 0 & \text{otherwise.} \end{cases}$$

Note that $X(t)$ can be obtained via monitoring loss in payoff or reduction in QoS, e.g. rate/video quality in WiFi application. We consider the following **distributed sensing algorithm** for updating the sensing probability at the start of the next frame $k+1$ for each player ℓ:

$$\begin{aligned} p_\ell(k+1) = \kappa(k) \max \Bigg\{ & p_{\min}, p_{\text{start}} \frac{1}{M} \sum_{t=1}^M (1 - \mathbf{1}_{\text{S}}(t)) \mathbf{1}_{\text{Sense}}(t) \\ & + \ p_\ell(k) \eta \exp^{-c_s} \exp^{-c_0 \hat{\mathcal{A}}(k)} \frac{1}{M} \sum_{t=1}^M \mathbf{1}_{\text{S}}(t) \mathbf{1}_{\text{Sense}}(t) \\ & + \ p_\ell(k) \frac{1}{M} \sum_{t=1}^M (1 - \mathbf{1}_{\text{Sense}}(t)) \Bigg\} \wedge 1 + (1 - \kappa(k)) p_\ell(k), \end{aligned} \tag{1}$$

with $x \wedge y = \min\{x, y\}$, and where $p_{\min} > 0, p_{\text{start}} > 0$ are minimum sensing probability and fixed reset sensing probability, respectively, c_0 and η are constants to be chosen later, and $\kappa(k)$ is the update step-size. It is important to note that this algorithm does not need the knowledge of the total number of other players $N - 1$ or their sensing strategies, and thus is easily implementable.

The second argument of the maximum in the sensing algorithm (1) contains three complementary terms (only one of them is non-zero for slot t in frame k), where the first represents the empirical measure with which the AP was found OFF on sensing scaled by a fixed reset sensing probability p_{start}, the second weighs the number of competing players (congestion penalty) and the cost of sensing exponentially with the existing sensing probability, a constant $\eta > 1$, and the empirical measure with which the AP was found ON on sensing, and the third introduces a damping factor that resists the change in sensing probability if sensing was not performed often enough in that frame.

The basic idea behind the update (1) is to significantly lower the sensing probability when there are a large number of other active players found in the current frame. This directly controls the congestion and incentivises sporadic sensing, and can be thought of as a backoff mechanism to implement a 'soft processor-sharing' routine.

On the other hand, if the number of other active players is low/moderate, and the empirical measure with which the AP was found ON on sensing (that tracks the connection rate λ_c of the server) is high, the sensing probability is increased to maximally utilise the opportunity provided by the server for service completions by each player. In a complementary sense, if the empirical measure with which the AP was found OFF on sensing is high, then the first term dominates and tries to lower the sensing probability. The sensing cost is also incorporated explicitly and weighed exponentially to limit the total sensing cost.

The following is the main result of the paper, showing that the update strategy (1), when followed by all N players, converges to a unique fixed point.

Theorem 2: If the following condition is satisfied

$$\frac{(N-1)c_0\ p_{\text{start}}\ \eta\frac{\lambda_c\lambda_d}{(\lambda_c+\lambda_d)^2}}{\left[1-\eta\frac{\lambda_d}{\lambda_c+\lambda_d}\right]^2}\left[\frac{1}{1-(1-\mu)(1-\lambda_c)}\right]\leq 1, \quad (2)$$

then the sensing update strategy (1) when followed by all N players converges to a *unique* fixed point, starting from any initial point. The unique fixed point also corresponds to a NE for a N player game, with individual utilities U_ℓ given by (5).

The left-hand side (LHS) of (2) is inherently a measure of congestion seen by each player; condition (2) specifies the congestion tolerance for the update algorithm that allows the convergence to a fixed point. Since c_0 and η are parameters under control, they can be chosen to satisfy (2) which determines the actual trajectory of the update strategy (1).

For proving Theorem 2, we first consider an *expected* version of the update strategy (1) and interpret that each player is updating that *expected* version so as to maximize some utility function for itself. This association is made only for theoretical purposes, and the actual update algorithm (1) does not need to know any statistics or expected values. Using that utility function, we define a game, for which there is a unique NE (under technical conditions) and to prove Theorem 2, show that the update strategy (1) converges to that NE. Next, we consider the *expected* version of (1) and develop the corresponding utility function and the game for the N players in Section II-C.

B. A steady-state version of the update rule

Instead of directly analysing the trajectory of the update rule (1), we first study an expected or steady state version of (1). To this end, observe that within a frame of large enough size M, the player sensing probabilities $\mathbf{p}\equiv(p_i)_{1\leq i\leq N}$, $p_i>0\ \forall i$ are fixed. It follows that, within a frame, the $\{0,1\}^N$-valued stochastic process that tracks where player $i, i=1,\dots,N$ is active (state 1) or not (state 0) at time slot $t=1,2,\dots$, is an irreducible and aperiodic discrete time Markov chain. By the ergodic theorem for discrete time Markov chains [18], the time average of the number of other players that are active during the time player ℓ is active, converges with probability 1 to the steady state expected number of active players in the server conditioned on tagged player ℓ being active, as the number of time slots M tend to ∞. For a fixed $\mathbf{p}$, let $\mathcal{A}(\mathbf{p})=\mathbb{E}\{\hat{A}|\mathbf{p}\}$ be the expected number of active players seen by the active tagged player including itself under steady state.

Lemma 3: For a fixed sensing probability vector $\mathbf{p}>0$, $\mathcal{A}(\mathbf{p})=\ 1+\sum_{j\in\Gamma_{-\ell}}\psi_j$, where ψ_j's are given by

$$\psi_j=\frac{p_j\lambda_c}{[1-(1-\mu)(1-\lambda_c)]}\frac{1}{[\lambda_c+p_j(1-\lambda_c)]}.^1 \quad (3)$$

Given $\mathbf{p}(k)$ at the beginning of frame k, and a sufficiently large frame size M, we replace $\hat{A}(k)$ by $\mathcal{A}(\mathbf{p}(k))=1+\sum_{j\in\Gamma_{-\ell}}\psi_j$ (Lemma 3), and all other random variables by their expected values in (1), to obtain the following 'expected' update equation:

$$\begin{aligned}p_\ell(k+1)\quad &=\kappa(k)\max\{p_{\min},p_{\text{start}}\frac{\lambda_d}{\lambda_c+\lambda_d}p_\ell(k),\\ &+\eta\exp^{-c_s}\frac{\lambda_c}{\lambda_c+\lambda_d}p_\ell(k)^2\prod_{j\in\Gamma_{-\ell}}\exp^{-c_0\psi_j}\\ &+p_\ell(k)(1-p_\ell(k))\}+(1-\kappa(k))p_\ell(k). \quad (4)\end{aligned}$$

since $\mathbb{E}\{(1-\mathbf{1}_{\text{S}}((1,k)))\mathbf{1}_{\text{Sense}}((1,k))|\mathbf{p}(k)\}=\frac{\lambda_d}{\lambda_c+\lambda_d}\mathbb{E}\{\mathbf{1}_{\text{Sense}}((1,k))|\mathbf{p}(k)\}$, and using the fact that AP ON/OFF mode is independent of sensing given $\mathbf{p}(k)$, $\mathbb{E}\{\mathbf{1}_{\text{Sense}}((1,k))|\mathbf{p}(k)\}=p_\ell(k)$.

C. Game Formulation

In this section, we adopt the view that each player updates its sensing strategy according to (4), so as to maximize some utility function for itself. Towards this end, consider a non-cooperative game $\mathcal{G}=\{N,p_\ell,U_\ell,\ell\in[1:N]\}$ with N players, utility function U_ℓ and strategy p_ℓ for player ℓ. The game $\mathcal{G}$ has a NE $\mathbf{p}^*$ if $U_\ell(p_\ell,\mathbf{p}^*_{-\ell})\leq U_\ell(p^*_\ell,\mathbf{p}^*_{-\ell}),\forall\ p_\ell\in[1:N]$. Next, in Theorem 4, we identify the utility function that each player is trying to selfishly maximize via update strategy (4).

Theorem 4: The utility function for player ℓ that (4) attempts to maximize is given by $U_\ell(\mathbf{p})=U_\ell(p_\ell,\mathbf{p}_{-\ell})=$

$$p_{\text{start}}\frac{\lambda_c}{\lambda_c+\lambda_d}\frac{p_\ell^2}{2}+\frac{p_\ell^3}{3}\left(\eta\exp^{-c_s}\frac{\lambda_d}{\lambda_c+\lambda_d}\prod_{i\in\Gamma_{-\ell}}e^{-c_0\psi_i}-1\right). \quad (5)$$

Moreover, if[2]

$$p_{\text{start}}\frac{\lambda_c}{\lambda_c+\lambda_d}+\eta\exp^{-c_s}\frac{\lambda_d}{\lambda_c+\lambda_d}<1, \quad (6)$$

then there exists a valid non-zero NE $\mathbf{p}^*$ for $\mathcal{G}$ that satisfies, $p^*_\ell=\frac{p_{\text{start}}\frac{\lambda_c}{\lambda_c+\lambda_d}}{\left[1-\eta\frac{\lambda_d}{\lambda_c+\lambda_d}\prod_{i\in\Gamma_{-\ell}}e^{-c_0\psi^*_i}\right]}$.

The proof is presented in Appendix A. The selfish utility function $U_\ell(p_\ell,\mathbf{p}_{-\ell})$ defined by (5) contains two terms that individually capture the natural benefit and cost for each user. The first term scales the sensing probability with the duty cycle (fraction of time the server is active) $\frac{\lambda_c}{\lambda_c+\lambda_d}$, so as to maximally utilize the server activity periods. The second term corresponds to the congestion (via ψ_i) and the sensing cost, and the utility decreases with the increasing number of competing players and the sensing cost.

Note that the utility function (5) has extra powers of p_ℓ's that are important for showing that there exists NE. Disregarding the quadratic and the cubic powers of p_ℓ, utility function (5) has all the required selfish properties, but the exact form that is amenable for analysis would have been difficult to guess to begin with. This approach helps in identifying the right utility function.

The best response strategy for player ℓ, assuming all other player strategies $\mathbf{p}_{-\ell}$ are fixed, is given by: $p_\ell^{\text{br}}=$

[1]The proof is elementary and omitted for lack of space.

[2]Condition (6) ensures that the updated value of probability is ≥ 0.

$\text{argmax}_{p_{\min}\le p_\ell} U_\ell(p_\ell, \mathbf{p}_{-\ell})$, which for game $\mathcal{G}$ (from $\frac{\partial U_\ell}{\partial p_\ell}=0$) evaluates to

$$p_\ell^{\text{br}} = \frac{p_{\text{start}}\frac{\lambda_c}{\lambda_c+\lambda_d}}{\left[1-\eta\frac{\lambda_d}{\lambda_c+\lambda_d}\prod_{i\in\Gamma_{-\ell}} e^{-c_0\psi_i}\right]}. \tag{7}$$

We next show, via a contraction mapping argument, that the NE for the game $\mathcal{G}$ is unique, and that the best response strategy (7) achieves it (Proof in Appendix B).

Theorem 5: If condition (2) holds, then the NE for the considered game $\mathcal{G}$ is unique, and the best response strategy (7) converges to the unique equilibrium.

Since the NE for the considered game $\mathcal{G}$ is unique, next, we use that to prove Theorem 2 by showing that the original update strategy (1) converges to the best response solution (7) for game $\mathcal{G}$.

D. Proof of Theorem 2

We are now ready to work towards proving Theorem 2. The first step in this direction is to reinterpret the expected update equation (4) in terms of a gradient descent algorithm for maximizing U_ℓ by player ℓ, given by

$$p_\ell(k+1) = \max\left\{p_{\min}, p_\ell(k) + \kappa\frac{\partial U_\ell(\mathbf{p}(k))}{\partial p_\ell}\right\}, \tag{8}$$

that is identical to the expected update equation (4), which is no surprise because of the definition of utility U_ℓ.

The first result we have with the gradient descent algorithm is its convergence to the NE depending on the step-size.

Lemma 6: Under the condition (6), with stepsize $\kappa \le 1$, the iterates of the gradient descent algorithm (8) converge to the best response solution (7) for player ℓ, under fixed $\mathbf{p}_{-\ell}$.

Thus, if all other players freeze their strategies $\mathbf{p}_{-\ell}$, then player ℓ can reach the best response to $\mathbf{p}_{-\ell}$ by running the gradient descent update equation (8) or the expected update strategy (4). Lemma 6 is applicable as long as each player's strategy is updated sequentially, which requires time dilation (i.e., each player updates its strategy not in every frame but after multiple frames depending on the convergence time) for converging to the best response solution, which eventually converges to the global NE as shown in Theorem 5.

The proof of Lemma 6 is provided in Appendix C, where we show that the utility function is β-smooth with $\beta = 2$, using which the convergence is established.

Finally, we now complete the proof of Theorem 2, by showing that the proposed update algorithm (1) converges to the best response strategy (7). Towards that end we make a correspondence between a stochastic sub-gradient algorithm and the update strategy (1) as follows.

E. A Stochastic Sub-Gradient interpretation of Update Strategy (1)

Let $v_\ell(k) = -p_\ell(k) + \frac{1}{M}\sum_{t=1}^{M} p_{\text{start}}(1-\mathbf{1}_{\text{S}}(t))\mathbf{1}_{\text{Sense}}(t)$

$$+\eta p_\ell(k)\exp^{-c_s}\exp^{-c_0\hat{A}(k)}\frac{1}{M}\sum_{t=1}^{M}\mathbf{1}_{\text{S}}(t)\mathbf{1}_{\text{Sense}}(t)$$
$$+\frac{p_\ell(k)}{M}\sum_{t=1}^{M}(1-\mathbf{1}_{\text{Sense}}(t)). \tag{9}$$

Then, (1) can be written as

$$p_\ell(k+1) = \max\{p_{\min}, p_\ell(k) + \kappa(k)v_\ell(k)\}, \tag{10}$$

where $\tilde{p}_{min} \ge \tilde{p}_{\min}$ will be chosen to satisfy technical condition required in Theorem 7. From the definition of utility function $U_\ell(\mathbf{p})$ (5), it is easy to check that $\mathbb{E}\{v_\ell|\mathbf{p}\} = \frac{\partial U_\ell(\mathbf{p})}{\partial p_\ell}$, and (1) or (10) is the *stochastic* gradient descent algorithm counterpart of (8). Thus, the update strategy (1) is solving a stochastic sub-gradient maximization of the utility function U_ℓ.

In a manner similar to Lemma 6, we next show that stochastic gradient descent algorithm (10) converges to the best response solution (7) for each player ℓ in the game $\mathcal{G}$ for fixed strategies $\mathbf{p}_{-\ell}$ under appropriate choice of step-size κ.

Theorem 7: With fixed $\mathbf{p}_{-\ell}$, for each player ℓ, the iterates of (10), converge to the best response solution (7) with probability 1 if the following conditions hold,

1) The step size $\kappa(k)$ satisfies $\kappa(k)\ge 0, \sum_{k=0}^{\infty}\kappa(k)=\infty$ and $\sum_{k=0}^{\infty}\kappa(k)^2<\infty$.
2) $p_{\min} \ge \tilde{p}_{\min} = \frac{p_{\text{start}}\lambda_c}{2[\lambda_c+\lambda_d-\lambda_c\eta e^{-c_0}]}$.

The proof is provided in Appendix D. With this, we have completed the proof of Theorem 2, since we have shown that the proposed sensing strategy (1) converges (if updated sequentially by each player) to the best response solution (7), which converges to the fixed point that corresponds to the unique NE of game $\mathcal{G}$ under condition (2).

Compared to best response strategy, the proposed sensing strategy (1) does not have precise knowledge of other nodes' sensing probabilities, but still converges to the same equilibrium point.

III. Numerical Results

We consider the WiFi network testbed model [19], where 31 APs are distributed uniformly randomly in an area of 2000 acres, with density $\rho = \frac{31}{2000\times4046}$ APs/m^2. $N = 5$, and each mobile's speed is $v = 30$ m/s in random orientation. The mobile is connected to an AP, if it is within $R = 250$ m from it. Under these settings, from [19], we have $\lambda_c = 2Rv\rho = 0.05745$ and $\lambda_d = \frac{v}{R} - 2Rv\rho = 0.06253$, and service rate $\mu = 5\lambda_c$. We let $p_{\text{start}} = 0.5$, then from (6), we need $\eta < 1.46$. For simulation, we consider $\eta = 1.2$. Moreover, to satisfy (2), we take $c_0 = 0.05$. In Fig. 1, we plot the trajectories of the sensing probabilities for any one player obtained by the best response updates, the gradient descent updates, and the actual learning (stochastic gradient descent) algorithm, and observe that the actual update is noisy but converge towards the equilibrium.

We also consider a twitter-like setup, where number of tweeting/competing users changes over time. We start with $N = 5$ users, and consider a time-horizon of $T = 100$ frames, where at $t = 0.33 * T^{th}$ frame, N increases to 100 and at $t = 0.67 * T^{th}$ frame, N goes back to 5. In Fig. 2, we plot the trajectories of the sensing probabilities with the three discussed algorithms with $\eta = 1.2$, $c_0 = 0.05$, $\lambda_c = 0.05$ and $\lambda_d = 0.06$, and $\mu = 5\lambda_c$. As expected, with increased N, the sensing probabilities converge to a lower value.

IV. Conclusions

In this paper, we considered competition models when there is uncertainty about the underlying resource availability, and there is competition from other users that are trying to extract maximum share of the available resource. Rather than directly considering a particular utility function, we instead started with an intuitive distributed adaptive strategy and showed that it converges to the NE of a sensing game with reasonable utility function for the studied problem. The approach presented in this paper is expected to be useful for many other related settings, e.g., uplink scheduling with quality of service guarantees, device-to-device communications etc.

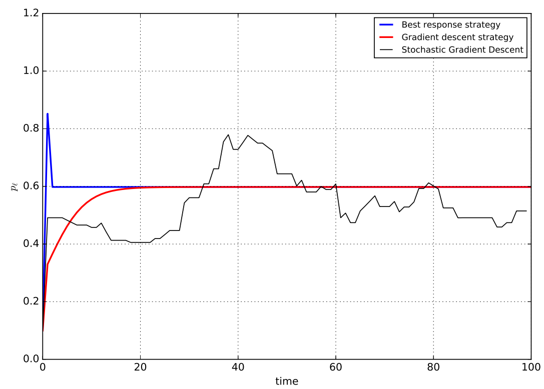

Fig. 1. The trajectories of the three update algorithms for the WiFi example.

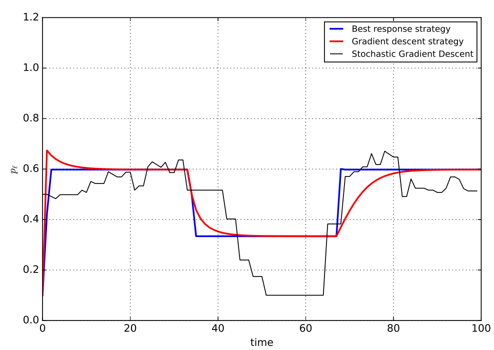

Fig. 2. Three update algorithm trajectories for the twitter example.

References

[1] W. E. Stein, A. Rapoport, D. A. Seale, H. Zhang, and R. Zwick, "Batch queues with choice of arrivals: Equilibrium analysis and experimental study," *Games and Economic Behavior*, vol. 59, no. 2, pp. 345 – 363, 2007. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0899825606001011

[2] H. Honnappa and R. Jain, "Strategic arrivals into queueing networks," in *2010 48th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, Sept 2010, pp. 820–827.

[3] M. Haviv, "When to arrive at a queue with tardiness costs?" *Performance evaluation*, vol. 70, no. 6, pp. 387–399, 2013.

[4] R. Jain, S. Juneja, and N. Shimkin, "The concert queueing game: to wait or to be late," *Discrete Event Dynamic Systems*, vol. 21, no. 1, pp. 103–138, 2011.

[5] E. Altman, P. Kumar, S. Venkatramanan, and A. Kumar, "Competition over timeline in social networks," in *Advances in Social Networks Analysis and Mining (ASONAM), 2013 IEEE/ACM International Conference on*. IEEE, 2013, pp. 1352–1357.

[6] A. Reiffers-Masson, E. Hargreaves, E. Altman, W. Caarls, and D. S. Menasche, "Timelines are publisher-driven caches: Analyzing and shaping timeline networks," in *NetEcon*, 2016.

[7] J. Jeong, Y. Yi, J.-w. Cho, S. Chong *et al.*, "Wi-fi sensing: Should mobiles sleep longer as they age?" in *INFOCOM, 2013 Proceedings IEEE*. IEEE, 2013, pp. 2328–2336.

[8] A. Kumar, S. R. B. Pillai, R. Vaze, and A. Gopalan, "Optimal wifi sensing via dynamic programming," in *Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt), 2015 13th International Symposium on*. IEEE, 2015, pp. 54–61.

[9] A. Tang, J.-W. Lee, J. Huang, M. Chiang, and A. R. Calderbank, "Reverse engineering MAC," in *2006 4th International Symposium on Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks*. IEEE, 2006, pp. 1–11.

[10] R. Kleinberg, G. Piliouras, and E. Tardos, "Multiplicative updates outperform generic no-regret learning in congestion games," in *Proceedings of the forty-first annual ACM symposium on Theory of computing*. ACM, 2009, pp. 533–542.

[11] W. Krichene, B. Drighès, and A. M. Bayen, "Online learning of nash equilibria in congestion games," *SIAM Journal on Control and Optimization*, vol. 53, no. 2, pp. 1056–1081, 2015.

[12] E. Friedman and S. Shenker, "Learning and implementation on the internet," *Manuscript. New Brunswick: Rutgers University, Department of Economics*, 1997.

[13] C. Daskalakis, R. Frongillo, C. H. Papadimitriou, G. Pierrakos, and G. Valiant, "On learning algorithms for nash equilibria," in *International Symposium on Algorithmic Game Theory*. Springer, 2010, pp. 114–125.

[14] Y. Shoham, R. Powers, and T. Grenager, "If multi-agent learning is the answer, what is the question?" *Artificial Intelligence*, vol. 171, no. 7, pp. 365–377, 2007.

[15] E. Altman and N. Shimkin, "Individual equilibrium and learning in processor sharing systems," *Operations Research*, vol. 46, no. 6, pp. 776–784, 1998.

[16] G. Kasbekar and A. Proutiere, "Opportunistic medium access in multichannel wireless systems: A learning approach," in *Communication, Control, and Computing (Allerton), 2010 48th Annual Allerton Conference on*. IEEE, 2010, pp. 1288–1294.

[17] X. Chen and J. Huang, "Distributed spectrum access with spatial reuse," *IEEE Journal on Selected Areas in Communications*, vol. 31, no. 3, pp. 593–603, 2013.

[18] R. Durrett, *Probability: theory and examples*. Cambridge university press, 2010.

[19] K.-H. Kim, A. W. Min, D. Gupta, P. Mohapatra, and J. P. Singh, "Improving energy efficiency of wi-fi sensing on smartphones," in *INFOCOM, 2011 Proceedings IEEE*. IEEE, 2011, pp. 2930–2938.

[20] M. J. Osborne and A. Rubinstein, *A course in game theory*. MIT press, 1994.

[21] R. Abraham, J. E. Marsden, and T. Ratiu, *Manifolds, tensor analysis, and applications*. Springer Science & Business Media, 2012, vol. 75.

[22] R. A. Horn and C. R. Johnson, *Matrix analysis*. Cambridge university press, 2012.

[23] Y. Nesterov, *Introductory lectures on convex optimization: A basic course*. Springer Science & Business Media, 2013, vol. 87.

[24] Y. Ermoliev and R. J.-B. Wets, *Numerical techniques for stochastic optimization problems*. International Institute for Applied Systems Analysis, 1984.

Appendix A
Proof of Theorem 4

In order to obtain a utility function corresponding to the expected update equation (4), consider the equilibrium point $\mathbf{p}^*$ for (4), with $p_{min} < p_\ell^* < 1, \ \forall \ \ell$, for which the update equation will satisfy the following fixed point equation,

$$\begin{aligned} p_\ell^* \quad &= \quad p_{\text{start}} \frac{\lambda_d}{\lambda_c + \lambda_d} p_\ell^* + p_\ell^*(1 - p_\ell^*) \\ &\quad + \eta \exp^{-c_s} \frac{\lambda_c}{\lambda_c + \lambda_d} (p_\ell^*)^2 \prod_{i \in \Gamma_{-\ell}} \exp^{-c_0 \psi_i^*}, \end{aligned} \quad (11)$$

where ψ_i^* is the function $\psi_i(p_i)$ evaluated when p_i^*. We need $0 \le p_\ell^* \le 1$, which is satisfied as long as (6) is satisfied.

Using (4), inherently each player is trying to maximize some utility function U_ℓ, and if at equilibrium (11) is satisfied, then one obvious choice of such utility function is that which satisfies $\frac{\partial U_\ell(\mathbf{p})}{\partial p_l} = 0$ at $\mathbf{p}^*$. Thus, by moving p_ℓ to RHS in (11), we have

$$\begin{aligned}\frac{\partial U_\ell(\mathbf{p})}{\partial p_\ell} \quad &= \quad p_{\text{start}} \frac{\lambda_d}{\lambda_c + \lambda_d} p_\ell + p_\ell(1-p_\ell) - p_\ell \\ &\quad + \eta \exp^{-c_s} \frac{\lambda_c}{\lambda_c + \lambda_d} p_\ell^2 \prod_{i \in \Gamma_{-\ell}} \exp^{-c_0 \psi_i},\end{aligned}$$

which gives the utility function (5) for player ℓ (unique upto to a constant), The set $\{p_\ell | p_{\min} \leq p_\ell \leq 1\}$ is a non-empty compact convex set on $\mathbb{R}$. Moreover, the utility function U_ℓ (5) is quasi-concave and continuous in p_ℓ (for lack of space we omit the proof here). Thus, using the Proposition 20.3 in [20] there exists a NE, where (11) is satisfied with ψ_i replaced by ψ_i^*.

Appendix B
Proof of Theorem 5

We use the following Theorem from [21] to prove this result.

Theorem 8: Let M be a complete metric space with metric d, and $f : M \to M$ be a mapping. Assume that there exists a constant γ such that $0 \leq \gamma < 1$ and $d(f(v), f(u)) \leq \gamma d(v,u)$ for all $u, v \in M$; such an f is called a contraction. Then f has a unique fixed point; that is, there exists a unique $u^* \in M$ such that $f(u^*) = u^*$. Furthermore, the sequence $u(t+1) = f(u(t))$ converges to this unique fixed point.

We will consider the best response strategy p_ℓ^{br} as the function f, and apply the above theorem with M being the Euclidean space $\mathbb{R}^N$ endowed with a norm $||\cdot||_2$. Let $d(\cdot)$ be the distance metric induced by this norm. Let $||\frac{\partial f}{\partial x}||$ be the Jacobian, then from properties of matrix norm [22], the following is true: $d(f(v), f(u)) = ||f(v) - f(u)|| \leq ||\frac{\partial f}{\partial x}|| ||u - v|| = ||\frac{\partial f}{\partial x}|| d(v,u)$, and for proving that f is a contraction mapping, it is sufficient to show that $||\frac{\partial f}{\partial x}|| < 1$ everywhere in x, and then invoke Theorem 8 to prove the claim.

Next, we work towards showing the infinity norm of $\mathbf{J}$ ($\mathbf{J}_{\ell j} \triangleq \frac{\partial p_\ell^{\text{br}}}{\partial p_j}$) is less than 1 which implies that $||\frac{\partial f}{\partial x}|| < 1$ for all players ℓ. By definition, $\mathbf{J}_{\ell j}$

$$= \begin{cases} \dfrac{-c_0 p_{\text{start}} \eta \exp^{-c_s} \frac{\lambda_c \lambda_d}{(\lambda_c+\lambda_d)^2} \left(\prod_{i\in\Gamma_{-\ell}} e^{-c_0\psi_i}\right) \sum_{k\in\Gamma_{-\ell}} \frac{\partial \psi_k}{\partial p_\ell}}{\left[1 - \eta \exp^{-c_s} \frac{\lambda_d}{\lambda_c+\lambda_d} \prod_{i\in\Gamma_{-\ell}} e^{-c_0\psi_i}\right]^2}, & \text{if } \ell = j, \\ \dfrac{-c_0 p_{\text{start}} \eta \exp^{-c_s} \frac{\lambda_c \lambda_d}{(\lambda_c+\lambda_d)^2} \left(\prod_{i\in\Gamma_{-\ell}} e^{-c_0\psi_i}\right)}{\left[1 - \eta \exp^{-c_s} \frac{\lambda_d}{\lambda_c+\lambda_d} \prod_{i\in\Gamma_{-\ell}} e^{-c_0\psi_i}\right]^2} \dfrac{\partial \psi_j}{\partial p_j}, & \text{o.w.} \end{cases}$$

From the definition of $\{\psi_k\}_{k\in\Gamma_{-\ell}}$ in (3), we have $\frac{\partial \psi_k}{\partial p_\ell} = 0 \quad \forall k \in \Gamma_{-\ell}$. Therefore, $\mathbf{J}_{\ell j} = 0$ for $\ell = j$. Next, we first bound the partial derivative $\frac{\partial \psi_j}{\partial p_j}$ by rewriting the definition of ψ_j from (3) as, $\psi_j = \frac{h_1 p_j}{h_2 + h_3 p_j}$, where $h_1 = \lambda_c$, $h_2 = [1-(1-\mu)(1-\lambda_c)]\lambda_c$, and $h_3 = (1-\lambda_c)[1-(1-\mu)(1-\lambda_c)]$. Taking the partial derivative of ψ_j w.r.t p_j, we get $\frac{\partial \psi_j}{\partial p_j} = \frac{h_1 h_2}{(h_2 + h_3 p_j)^2}$, which trivially imply the following bounds, $\frac{h_1 h_2}{(h_2+h_3)^2} \leq \frac{\partial \psi_j}{\partial p_j} \leq \frac{h_1}{h_2}$. Using the expressions for h_i, we get the following bounds

$$\frac{\lambda_c^2}{[1-(1-\mu)(1-\lambda_c)]} \leq \frac{\partial \psi_j}{\partial p_j} \leq \frac{1}{[1-(1-\mu)(1-\lambda_c)]}. \quad (12)$$

We now upper bound $||\mathbf{J}||_\infty = \max_\ell \sum_{j=1}^N |J_{\ell j}|$, as follows. Since $J_{\ell,\ell} = 0$, and $J_{\ell,j} < 0$, we have $||\mathbf{J}||_\infty$

$$\begin{aligned} &= \max_\ell \left\{ \sum_{j\in\Gamma_{-\ell}} \frac{\frac{c_0\ p_{\text{start}}\ \eta \exp^{-c_s}\ \lambda_c \lambda_d}{(\lambda_c+\lambda_d)^2} \left(\prod_{i\in\Gamma_{-\ell}} e^{-c_0\psi_i}\right) \left[\frac{\partial \psi_j}{\partial p_j}\right]}{\left[1 - \eta \exp^{-c_s}\ \frac{\lambda_d}{\lambda_c+\lambda_d} \prod_{i\in\Gamma_{-\ell}} e^{-c_0\psi_i}\right]^2} \right\}, \\ &\overset{(a)}{\leq} \max_\ell \left\{ \sum_{j\in\Gamma_{-\ell}} \frac{c_0\ p_{\text{start}}\ \eta \exp^{-c_s}\ \frac{\lambda_c\lambda_d}{(\lambda_c+\lambda_d)^2}}{\left[1 - \eta \exp^{-c_s}\ \frac{\lambda_d}{\lambda_c+\lambda_d}\right]^2} \left[\frac{\partial \psi_j}{\partial p_j}\right] \right\}, \\ &\overset{(b)}{\leq} \max_\ell \left\{ |\Gamma_{-\ell}| \frac{c_0\ p_{\text{start}}\ \eta \exp^{-c_s}\ \frac{\lambda_c\lambda_d}{(\lambda_c+\lambda_d)^2}}{\left[1 - \eta \exp^{-c_s}\ \frac{\lambda_d}{\lambda_c+\lambda_d}\right]^2} \left[\frac{\partial \psi_j}{\partial p_j}\right] \right\}, \\ &\overset{(c)}{\leq} \max_\ell \left\{ \frac{(N-1) c_0\ p_{\text{start}}\ \eta \exp^{-c_s}\ \frac{\lambda_c\lambda_d}{(\lambda_c+\lambda_d)^2}}{\left[1 - \eta \exp^{-c_s}\ \frac{\lambda_d}{\lambda_c+\lambda_d}\right]^2} \left[\frac{\partial \psi_j}{\partial p_j}\right] \right\}, \\ &\overset{(d)}{\leq} \max_\ell \left\{ \frac{(N-1) c_0\ p_{\text{start}}\ \eta \exp^{-c_s}\ \frac{\lambda_c\lambda_d}{(\lambda_c+\lambda_d)^2}}{\left[1 - \eta \exp^{-c_s}\ \frac{\lambda_d}{\lambda_c+\lambda_d}\right]^2} \right. \\ &\qquad \left. \left[\frac{1}{1-(1-\mu)(1-\lambda_c)}\right] \right\}, \end{aligned} \quad (13)$$

where (a) follows since $\prod_{i\in\Gamma_{-\ell}} e^{-c_0\psi_i} \leq 1$, (b) follows by replacing the outer sum by $\Gamma_{-\ell}$, (c) follows since $|\Gamma_{-\ell}| = N-1$, and in (d) we use the upper bound (12).

Note that the argument of the max in (13) does not depend on ℓ, hence to make $||\mathbf{J}||_\infty \leq 1$ it is sufficient for the argument of the max to be less than 1, i.e.

$$\frac{(N-1) c_0\ p_{\text{start}}\ \eta \exp^{-c_s}\ \frac{\lambda_c\lambda_d}{(\lambda_c+\lambda_d)^2} \left[\frac{1}{1-(1-\mu)(1-\lambda_c)}\right]}{\left[1 - \eta \exp^{-c_s}\ \frac{\lambda_d}{\lambda_c+\lambda_d}\right]^2} < 1.$$

This inequality is satisfied by choosing appropriately the values of parameters η and c_0 as specified in condition (2).

Appendix C
Proof of Lemma 6

Definition 9: A function $f : \mathbb{R} \to \mathbb{R}$ is Lipschitz continuous with constant $L < \infty$ if $||f(x) - f(y)|| \leq L||x - y|| \ \forall\ x, y$. Moreover, a function f whose derivative is Lipschitz continuous with constant $\beta < \infty$, i.e., $||\nabla f(x) - \nabla f(y)|| \leq \beta ||x - y|| \ \forall\ x, y$, is called a β-smooth function.

We will use the following Theorem to prove Lemma 6.

Theorem 10: [23, §1.2.3] Let f be a β smooth function and $f^* = \min f(x) > -\infty$. Then the gradient descent algorithm with a constant step size $\kappa < \frac{2}{\beta}$ converges to a stationary point i.e., the set $\{x : \nabla f(x) = 0\}$

For fixed $\mathbf{p}_{-\ell}$, let player ℓ update the strategy using the gradient descent algorithm, and let $f \equiv U_\ell$. For player ℓ, the gradient expression is given by,

$$\nabla f = \frac{\partial U_\ell}{\partial p_\ell} = p_{\text{start}} \frac{\lambda_c}{\lambda_c+\lambda_d} p_\ell + p_\ell(1-p_\ell) - p_\ell + \eta \exp^{-c_s} \frac{\lambda_d}{\lambda_c+\lambda_d} p_\ell^2 \prod_{i\in\Gamma_{-\ell}} e^{-c_0\psi_i}. \quad (14)$$

Now we check the β-smoothness condition for f, and find a bound on the constant as follows. For that purpose,

$$\begin{aligned}
\left| \left|\frac{\partial U_\ell}{\partial p_\ell}\right|_{p_\ell=x} - \left|\frac{\partial U_\ell}{\partial p_\ell}\right|_{p_\ell=y} \right| &= \Bigg| p_{\text{start}} \frac{\lambda_c}{\lambda_c+\lambda_d}(x-y) - x^2 + y^2 \\
&\quad + \eta \exp^{-c_s} \frac{\lambda_d}{\lambda_c+\lambda_d}(x^2-y^2) \prod_{i\in\Gamma_{-\ell}} e^{-c_0\psi_i} \Bigg|, \\
&\overset{(a)}{\le} |x-y| \Bigg| p_{\text{start}} \frac{\lambda_c}{\lambda_c+\lambda_d} - (x+y) \\
&\quad + \eta \exp^{-c_s} \frac{\lambda_d}{\lambda_c+\lambda_d}(x+y) \prod_{i\in\Gamma_{-\ell}} e^{-c_0\psi_i} \Bigg|, \\
&\overset{(b)}{\le} |x-y| \Bigg| p_{\text{start}} \frac{\lambda_c}{\lambda_c+\lambda_d} \\
&\quad - \left(1 - \eta \exp^{-c_s} \frac{\lambda_d}{\lambda_c+\lambda_d} \prod_{i\in\Gamma_{-\ell}} e^{-c_0\psi_i}\right)(x+y) \Bigg|, \\
&\overset{(c)}{\le} ||(x,\mathbf{p}_{-\ell}) - (y,\mathbf{p}_{-\ell})|| \Bigg| p_{\text{start}} \frac{\lambda_c}{\lambda_c+\lambda_d} \\
&\quad - \left(1 - \eta \exp^{-c_s} \frac{\lambda_d}{\lambda_c+\lambda_d} \prod_{i\in\Gamma_{-\ell}} e^{-c_0\psi_i}\right)(x+y) \Bigg| \quad (15)
\end{aligned}$$

where (a) follows since $|pq| \le |p||q| \ \forall p,q \in \mathbb{R}$, (b) involves rearrangement of terms, and (c) follows since $|x-y| \le ||(x,\mathbf{p}_{-\ell}) - (y,\mathbf{p}_{-\ell})||$, where $(x,\mathbf{p}_{-\ell})$ is the N-length vector. If (6) holds, then each of the two terms of (6) are < 1, namely: $0 < \eta \exp^{-c_s} \frac{\lambda_d}{\lambda_c+\lambda_d} < 1$, and $0 < p_{\text{start}} \frac{\lambda_c}{\lambda_c+\lambda_d} < 1$. Moreover, we have $\prod_{i\in\Gamma_{-\ell}} e^{-c_0\psi_i} > 0$, $x \le 1$, and $y \le 1$. Using these inequalities, the second term in right-hand side of (15), can be bounded by 2 and we get

$$\left| \left|\frac{\partial U_\ell}{\partial p_\ell}\right|_{p_\ell=x} - \left|\frac{\partial U_\ell}{\partial p_\ell}\right|_{p_\ell=y} \right| < 2||(x,\mathbf{p}_{-\ell}) - (y,\mathbf{p}_{-\ell})||. \quad (16)$$

Thus, U_ℓ is β-smooth with $\beta < 2$. Moreover, since $\beta < 2$, $\exists$ a $\epsilon > 0$ such that, $\frac{2}{\beta} > 1+\epsilon$. Thus, from Theorem 10, if stepsize κ satisfies, $\kappa < 1+\epsilon$, then the iterates of the gradient descent algorithm (8) converge to the stationary point. For fixed $\mathbf{p}_{-\ell}$, the stationary point is the best response solution, and hence we have the result.

Appendix D
Proof of Theorem 7

Theorem 11: [Theorem 6.2 [24]] Consider

$$\max_{x\in[a,b]} F(x), \quad (17)$$

where F is a concave, continuous one-dimensional function, and let X^* be the set of optimal solutions. Consider the following stochastic subgradient projection method to solve (17): $x(t+1) = \max\{a, \min\{b, x(t) + s(t)\xi(t)\}\}, t = 0,1,2,\ldots$ If the following conditions are satisfied

1) $F(x^*) - F(x(t)) \le \mathbb{E}[\xi(t)|x(0),\ldots,x(t)]\{x^* - x(t)\} + \gamma_0(t)$, where $\gamma_0(t)$ may depend on $(x(0),\ldots,x(t)), x^* \in X^*$,
2) step size $s(t)$, $s(t) \ge 0, \sum_{t=0}^{\infty} s(t) = \infty$, and
3) $\sum_{t=0}^{\infty} \mathbb{E}\left[s(t)|\gamma_0(t)| + s^2(t)|\xi(t)|^2\right] < \infty$,

Then $\lim_{t\to\infty} x(t) \in X^*$ with probability 1.

We will use Theorem 11 to prove Theorem 7. From (10), the proposed learning algorithm is

$$p_\ell(t+1) = \max\{p_{\min}, p_\ell(t) + \kappa(t)v_\ell(t)\}, \quad (18)$$

where we will choose $p_{\min} \ge \tilde{p}_{\min}$ to satisfy conditions of Theorem 11. We first note that $U_\ell(\mathbf{p})$ is concave for $p_\ell \in [p_\ell^1, 1]$ from Lemma 12, where $p_\ell^1 = \frac{p_{\text{start}}\lambda_c}{2\left[\lambda_c+\lambda_d\left[1-\eta\prod_{i\in\Gamma_{-\ell}} e^{-c_0\psi_i}\right]\right]}$. Thereafter, in order to apply the Theorem 11 for our system, make use of the following mappings: $p_\ell \leftrightarrow x$, $U_\ell(p_\ell) \leftrightarrow F(x)$, $\tilde{p}_{\min} \leftrightarrow a$, $1 \leftrightarrow b$, $\{p_\ell^{\text{br}}\} \leftrightarrow X^*$, and $v_\ell(t) \leftrightarrow \xi(t)$. Also note that the best response strategy p_ℓ^{br} (7), satisfies $p_\ell^{\text{br}} = 2p_\ell^1$. Thus, we choose $\tilde{p}_{\min} = \max\{p_{\min}, p_\ell^1\}$.

Recall that for fixed $\mathbf{p}_{-\ell}$ (strategy of all other players), the maximizer of the utility function U_ℓ is p_ℓ^{br}, to which we want the update equation (18) to converge. Thus, to use Theorem 11 we need to need to ensure that p_ℓ^{br} does indeed lie in the range $[a,b] = [\tilde{p}_{\min}, 1]$. In order to satisfy $p_\ell^{\text{br}} \in [\tilde{p}_{\min}, 1]$, we need both $p_{\min} \le p_\ell^{\text{br}}$ and $p_\ell^1 \le p_\ell^{\text{br}}$, where the latter is automatically satisfied since $p_\ell^{\text{br}} = 2p_\ell^1$, and the former because of condition 2 in our theorem statement. Thus, we have $\max\{p_{\min}, p_\ell^1\} = \tilde{p}_{\min} \le p_\ell^{\text{br}} \le 1 \quad \forall \ell \in \Gamma_{-\ell}$.

Since the utility function $U_\ell(p_\ell)$ is strictly concave within the range $p_\ell \in [\tilde{p}_{\min}, 1]$ (Lemma 12) and $\mathbb{E}\{v_\ell|\mathbf{p}\} = \frac{\partial U_\ell(\mathbf{p})}{\partial p_\ell}$, first condition of Theorem 11 holds with $\gamma_0(t) = 0$. Moreover, as we have diminishing step-size $\kappa(k)$ which satisfies condition 1 of our theorem statement, and $|v(t)| \le (\eta + p_{\text{start}})$ (easy to see from (9)). Hence, all the required conditions for Theorem 11 are satisfied, and we conclude that $p_\ell(t)$ following (18) converges to the best response strategy p_ℓ^{br} with probability 1.

Lemma 12: The utility function U_ℓ is concave in p_ℓ for $p_\ell \in [p_\ell^1, 1]$, where $p_\ell^1 = \frac{p_{\text{start}}\lambda_c}{2\left[\lambda_c+\lambda_d\left[1-\eta\prod_{i\in\Gamma_{-\ell}} e^{-c_0\psi_i}\right]\right]}$.

Proof: Note that $\frac{\partial^2 U_\ell}{\partial p_\ell \partial p_k}$

$$= \begin{cases} p_{\text{start}} \frac{\lambda_c}{\lambda_c+\lambda_d} - 2p_\ell\left[1 - \eta\frac{\lambda_d}{\lambda_c+\lambda_d}\left(\prod_{i\in\Gamma_{-\ell}} e^{-c_0\psi_i}\right)\right], & \text{for } k = \ell, \\ -c_0\eta\frac{\lambda_d}{\lambda_c+\lambda_d} p_\ell^2 \prod_{i\in\Gamma_{-\ell}} e^{-c_0\psi_i}\frac{\partial\psi_k}{\partial p_k}, & \text{o.w.} \end{cases} \quad (19)$$

Thus, for $p_\ell \in [p_\ell^1, 1]$, $\frac{\partial^2 U_\ell}{\partial p_\ell^2} \le 0$. □