A Game-Theoretic Perspective on Advance Reservations

Eran Simhon and David Starobinski

Abstract

Advance reservation is a fundamental paradigm for resource allocation. It is employed in various economic sectors, including cloud computing and communication networks. Although advance reservations are widespread, little is known about the strategic behavior of users facing the decision whether to reserve a resource in advance or not. In this article, we present a game-theoretic framework, called Advance Reservation (AR) games, to analyze this strategic behavior. We use AR games to analyze the impact of pricing, charging, and information sharing policies on the economic equilibria of the system and on its dynamic behavior. The analysis yields several insights on how a service provider should design a system that supports advance reservations.

Introduction

Advance reservation (AR) services form a pillar of the economy. They are widely deployed in the industries of transportation (e.g., for reserving airplane and train tickets), lodging (e.g., for booking hotel rooms), and health care (e.g., for scheduling medical appointments). AR is also gaining popularity in communication networks [1, 2] and cloud computing [3, 4, 5].

A large portion of the existing research of advance reservations in communication networks focuses on algorithmic aspects, such as scheduling and routing [6, 7, 8]. Yet, in services supporting AR, it is often up to the users to decide whether to make a reservation in advance. Hence, understanding the strategic behavior of users in systems that support AR is a fundamental problem.

In this article, we describe a game-theoretic framework, called *Advance Reservation (AR) games*, that helps in reasoning about the strategic behavior of users in systems supporting advance reservations. In this framework, a random set of players, arriving at different times, request to be served in a specific time slot. Users can either make reservations in advance, at a certain cost, or defer their decisions and request service on the spot.

The AR cost may be a fee set by the service provider, the time or resources required for making the reservation, or the cost of financing payment. When a user avoids AR, this cost is spared. However, the probability to get service in the desired time slot is reduced.

Using AR games, we analyze the impact of pricing and other policies set by the provider on the behavior of the users, and in turn, on the economic outcomes of the system.

First, we investigate the impact of *pricing policies*. Our analysis reveals the different types of game equilibria. We show that, in many cases, the AR fee that brings the maximum possible revenue to the provider has other equilibria, including one yielding no revenue. We then explain how the provider can set the AR fee to guarantee a positive revenue from advance reservations.

Next, we use AR games to evaluate the impact of *charging policies*. We investigate whether it is worthwhile for a provider to charge a fee from all users making advance reservation or only from users that get service.

Next, we consider the impact of *information sharing policies*. We use AR games to answer whether it is to the provider's interest to inform users about the

The authors are with Boston University.

This research was supported in part by the US National Science Foundation under grant CNS-1117160.

number of servers left available, or hide that information from them.

Finally, we analyze *dynamic AR games*. The analysis sheds light into whether the system converges to an equilibrium, and if yes to which, or cycles.

Details of the results presented in this article can be found in [9, 10, 11]. Note that the operations research literature contains several results on the management of advance reservations. For instance, the work in [12] considers admission control strategies in reservation system with different classes of customers, while [13] deals with policies for accepting or rejecting restaurant reservations, and [14] analyzes the effects of customer regrets. None of this prior work considers the *strategic* behavior of customers in making AR, namely, that decisions of customers are not only influenced by prices and policies set by providers but also by their beliefs about the decisions of other customers.

AR Games

In this section, we present the framework of AR games. Table 1 summarizes the notations used throughout the article.

AR games are a type of non-cooperative game. A non-cooperative game consists of a set of players, in which each player follows a strategy that determines what action she chooses in any situation that she faces. Each player is associated with a payoff function, which describes her payoff as a function of her chosen action and the actions of all other players.

Many games studied in the literature assume that the number of players is fixed and known by everyone. However, when deciding whether to reserve resources in advance, users typically do not know how many other users compete for the same set of resources. Therefore, a crucial element of AR games is to consider a random number of players.

In AR games, the system consists of N servers that offer a time-slotted service. This model can be used to represent various practical systems (e.g., a link with a fixed number of circuits or a cloud cluster with a fixed number of virtual machines). We define the *demand* D to be the number of users requesting service at a given time slot (each user requests one server). The random variable D takes integer values and is independent of the history. It follows a general probability distribution with mean λ.

Notation	Description
N	Number of servers.
D	The demand, a random variable that represents the number of users requesting service.
λ	The mean demand.
C	Reservation cost.
$\sigma(t)$	Strategy function. The input is the lead time t and the output is an action (AR or not AR).
$D_{AR}(\tau)$	A random variable representing the number of AR requests, under a strategy with threshold τ.
$\pi_{AR}(\tau)$	Expected payoff function. The input is the threshold τ. The output is the probability that a user with lead time τ gets service if she makes AR.
$\pi_{AR'}(\tau)$	Expected payoff function. The input is the threshold τ. The output is the probability that a user with lead time τ gets service if she does not make AR.
$C(\tau)$	Cost function. The input is the threshold τ. The output is the reservation cost that leads to that threshold.
$R(\tau)$	Revenue function. The input is the threshold τ. The output is the revenue from the AR fee.
$\sigma(t,n)$	A strategy function. The input is the lead time t and the number of available servers n. The output is an action.
$\pi_n(\tau_n)$	Expected payoff function. The input is the threshold τ_n. The output is the probability that a user with lead time τ_n gets service, if she does not make AR and n servers are available.
$\hat{\alpha}_i$	An estimator of threshold followed at the i-th iteration of a dynamic game.

Table 1: Notation summary

For each user, the time elapsing between her arrival (i.e., the point at which she realizes that service will be needed in a future slot) and the slot starting time is referred to as her *lead time*. The lead time is a continuous positive random variable following an arbitrary probability distribution. As an example, consider a system with several servers where each slot lasts for one day, starting at 12:00 AM. A user realizes on Tuesday at 9:00 PM that she will need service

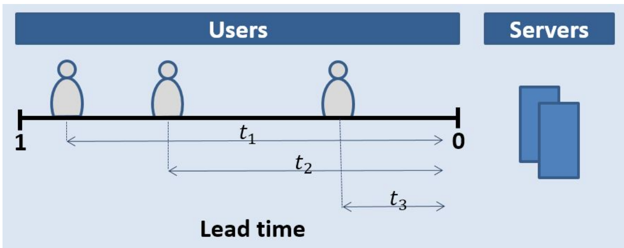

Figure 1: Example of a realization of the demand (i.e., the number of users requesting service) and the lead times in a system with two servers. Users with greater lead times have the opportunity to reserve a server earlier. Note that the realizations of the demand and lead times in different slots are independent.

on Friday. Then, her lead time is 51 hours.

Any lead time distribution can be converted into a continuous uniform distribution in $[0, 1]$, using a probability integral transformation [9]. Hence, from now and on, we will assume that the lead time is a uniform random variable in $[0, 1]$, denoted by T. Figure 1 illustrates the model.

Users have no prior information about the availability of servers (later in the article, we study a variant of the game where information about server availability is shared with the users). Each user decides whether to make AR or not (respectively denoted by AR and AR'), based on her own lead time and on statistical information about other users (i.e., the distributions of the demand and the lead time).

The servers are allocated in a *first-reserved-first-allocated* fashion. If $D > N$, but the number of reservations is smaller than N, then the unreserved servers are arbitrarily allocated among the users that requested service but did not make AR.

All the users have the same utility U from service. Without loss of generality, we set $U = 1$. Making AR is associated with a fixed cost $C < 1$. Hence, the payoff of a user that gets service is $1 - C$ if she chooses the action AR and 1 if she chooses the action AR'. The AR cost C reflects all aspects of making reservation. Analyzing a game with negative cost is trivial (all users make AR). Thus, we assume, henceforth, that $C > 0$.

A user that does not get service in her desired slot leaves the system with zero payoff, regardless of whether she attempted to make AR or not (later in the article, we study a variant of the game where AR fees are charged in advance from all users attempting AR).

Since the random realizations of the demand and lead times for each slot are independent of those in other slots, it is sufficient to analyze the game in one slot.

Equilibria Analysis

In a non-cooperative game, each user individually chooses a strategy that maximizes her expected payoff. A Nash equilibrium is a balanced state, in which none of the users has any incentive to deviate from her chosen strategy after observing the strategies chosen by the other users. Naturally, equilibrium strategies are those of interest. Since all users are statistically identical, we only consider symmetric equilibria (a common assumption made in the analysis of queueing games [15]). In a symmetric equilibrium, all users follow the same joint strategy function.

Since users only differ by their lead times, we can define a strategy function σ, which maps a lead time t to an action AR or AR'. As we show next, at equilibrium, the strategy function must be a threshold function of the form:

$$\sigma(t) = \begin{cases} \text{AR} & \text{if } t > \tau, \\ \text{AR'} & \text{if } t \le \tau, \end{cases} \quad (1)$$

where $\tau \in [0, 1]$ is the value of the threshold. Hence, under a threshold strategy, only users whose lead times are greater than the threshold τ make AR (i.e., only users who arrive early enough). If all users follow a strategy with threshold τ, then τ also represents the expected fraction of users that do not make AR.

All equilibrium strategies must be of threshold form for the following reason. If a user makes AR, the probability that she gets service is a non-decreasing function of her lead time. If she does not make AR, the probability to get service does not depend on her lead time. Thus, we have the following property: given any strategy function followed by all users, if a given user is better off making AR, then all users with greater lead time are also better off making AR. Similarly, if a given user is better off not making AR, then all users with smaller lead time are also better off not making AR.

We distinguish between two types of threshold equilibrium. In the first type, the threshold is $\tau = 1$, which means that none of the users is making AR. We refer to this equilibrium as *none-make-AR*. In the second type, the threshold is $\tau \in (0, 1)$, which means that the number of users making AR is a random variable. We refer to this equilibrium as *some-make-AR*. An equilibrium where all users make AR, regardless of their lead times, does not exist [9].

We next explain the procedure to establish the equilibrium threshold strategies and their different types. For any threshold $\tau \in (0, 1)$, consider a virtual user whose lead time is exactly equal to the threshold. We refer to this virtual user as a *threshold user*. Denote the probability that a threshold user gets service upon making an advance reservation by $\pi_{AR}(\tau)$ and the probability that a threshold user gets service upon not making an advance reservation by $\pi_{AR'}(\tau)$. Both functions are continuous (the formulas of $\pi_{AR}(\tau)$ and $\pi_{AR'}(\tau)$ can be found in [9]).

In a *some-make-AR* equilibrium, the threshold user must be indifferent between the actions AR and AR'. Thus, a strategy with threshold τ is a *some-make-AR* equilibrium if and only if:

$$(1 - C)\pi_{AR}(\tau) = \pi_{AR'}(\tau). \qquad (2)$$

By isolating C in the equation above, we can define a function that determines for any given threshold τ, what cost leads to that threshold. We denote this function by $C(\tau)$.

By studying the structure of the function $C(\tau)$, we can prove that the game has the following structure:

- **Low AR costs** only yield *some-make-AR* equilibria.
- **Medium AR costs** yield several equilibria, which include both *some-make-AR* equilibria and a *none-make-AR* equilibrium.
- **High AR costs** only yield a *none-make-AR* equilibrium.

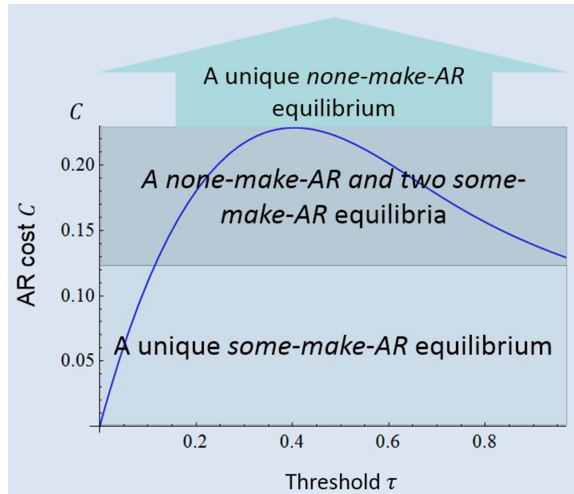

Figure 2: The cost C as a function of the threshold τ, which represents the expected fraction of users that do not make AR under a *some-make-AR equilibrium strategy*. If the AR cost exceeds $C(1)$, then a *none-make-AR equilibrium* also exists. The cost function reveals the equilibrium structure of the game. The parameters are $N = 10$ servers and a demand following a Poisson distribution with mean $\lambda = 10$.

Figure 2 shows the cost function $C(\tau)$ in a game with $N = 10$ servers and a Poisson distributed demand with mean $\lambda = 10$. As shown in the figure, for any AR cost falling in the lower range, there is one corresponding threshold value and, therefore, one *some-make-AR* equilibrium. For any AR cost falling in the middle range, there are two corresponding threshold values and, therefore, two *some-make-AR* equilibria. In addition, *none-make-AR* is also an equilibrium. Finally, AR costs belonging to the upper range (i.e., costs greater than the maximum value of $C(\tau)$) only yield a *none-make-AR* equilibrium.

Impact of Pricing

Suppose the AR cost is a fee charged by the provider. AR games reveal an interesting dilemma that a provider may face when attempting to maximize his revenue under strategic customer behavior.

Let $D_{AR}(\tau)$ be a random variable denoting the number of reservation requests under a threshold strategy τ. The expected revenue as a function of τ is

$$R(\tau) = C(\tau) \cdot \mathbb{E}[\min\{D_{AR}(\tau), N\}]. \tag{3}$$

The reason for taking the minimum between N and $D_{AR}(\tau)$ is that users do not have information about service availability. Thus, the number of reservation requests may exceed the number of servers. However, fees are only charged from users getting service.

Given the parameters of the system, one can find which value of τ maximizes $R(\tau)$. Numerical analysis shows that the fee that leads to the optimal threshold often belongs to the range of medium AR costs. By choosing a fee in that range, the provider takes the risk of ending up with zero revenue.

As an example, consider again a system with $N = 10$ servers and a demand that follows a Poisson distribution with $\lambda = 10$. For these settings, the maximum value of $R(\tau)$ is 1.5. This maximum is achieved with the threshold $\tau = 0.29$ (i.e., on average, 71 percent of the users make AR). The fee corresponding to this threshold is $C = 0.215$. Figure 2 shows that this fee yields multiple equilibria including a *none-make-AR* one.

Instead of taking the risk of ending up with zero revenue, the provider may opt to be risk-averse and to set a fee yielding a sub-optimal, but guaranteed revenue. As shown in Figure 3, the highest fee that yields a unique *some-make-AR* equilibrium is $C = 0.125$. At this equilibrium, on average, 88 percent of the users make AR and the expected revenue is 1.017. This revenue is about 30% lower than the optimal revenue.

Impact of Charging Policy

The model of the previous section assumes that AR fees are charged only from users granted service (we refer to that charging scheme as the first policy and the resulting game as the first game). In this section, we consider a scheme in which all users that make AR requests pay the fee, even if not granted service (we refer to this charging scheme as the second policy and the resulting game as the second game). Our goal is to evaluate which policy is more profitable for the provider.

We denote the cost and revenue functions of policy i, where $i \in \{1, 2\}$, by $C_i(\tau)$ and $R_i(\tau)$, respectively. Under the second policy, the payoff of users that make AR and do not get service is $-C$ rather than 0, as in the first policy.

Using similar arguments as in the analysis of the first policy, one can show that this game has an equilibrium structure with three ranges of costs. The types of equilibria in each range are the same as in the first game. The difference between the two games is the cost function. In the second game, a *some-make-AR* equilibrium exists if and only if:

$$\pi_{AR}(\tau) - C = \pi_{AR'}(\tau). \tag{4}$$

As before, we isolate C and define a cost function $C_2(\tau)$. From the definition of the two cost functions, one can show that $C_1(\tau) > C_2(\tau)$, for any $\tau \in (0, 1)$. This result has an intuitive interpretation. Under the second policy, the expected payoff of a user that makes AR is smaller. Hence, the provider must charge a smaller fee, in order to convince the same fraction of users to make AR.

The provider revenue from AR fees, under the second policy, is simply the number of users making AR requests multiplied by the reservation fee. Thus, the expected revenue is:

$$R_2(\tau) = C_2(\tau) \cdot \mathbb{E}[D_{AR}(\tau)]. \tag{5}$$

For any given threshold τ, the expected number of users paying the AR fee is obviously higher under the second policy. However, since $C_1(\tau) > C_2(\tau)$, there exists a trade-off between the two policies.

One can show that for any threshold τ and any distribution of D, the first policy always yields higher revenue than the second one, that is,

$$R_1(\tau) > R_2(\tau). \tag{6}$$

In other words, charging a fee from all users attempting to make AR, including those not granted service, is never optimal [9].

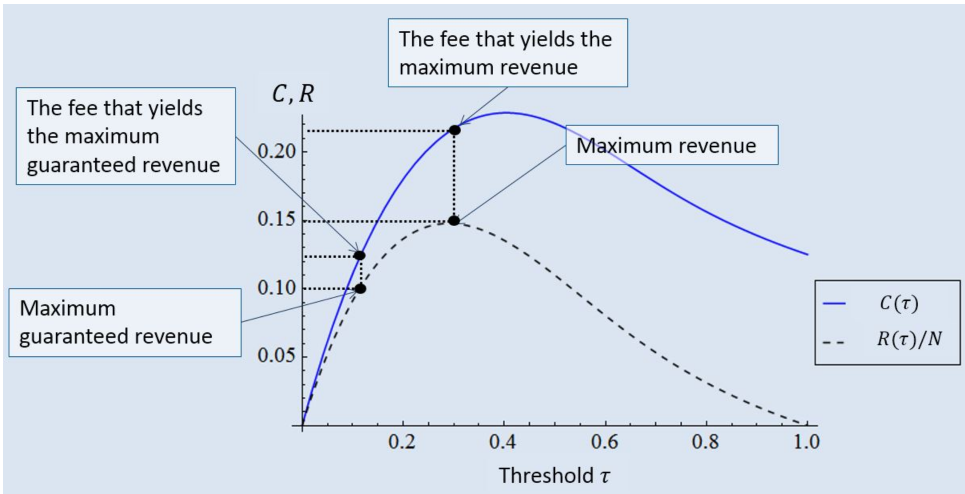

Figure 3: The fee and the average revenue per server as a function of the threshold, for the same system as in Figure 2. The fee that leads to the equilibrium with the greatest revenue, also leads to an equilibrium with zero revenue. By lowering the fee, a provider can guarantee a positive revenue, though smaller than optimal.

Impact of Information Sharing

In this section, we use the framework of AR games to explore the impact of sharing information about service availability with the users. Our goal is to determine whether a provider, wishing to maximize the number of reservations, is better off sharing information about service availability or not. Towards that end, we define and analyze a variant of the game, in which the number of available servers $n \in \{0, 1, \ldots, N\}$, is shared with the users upon their arrivals.

When information about availability of servers is shared, all users that make AR (after noticing that at least one server is available) have a fixed payoff $1 - C$. The expected payoff of a user that does not make AR is equal to her probability to get service, conditioned on the number of available servers she observes upon her arrival.

Using similar argument as in the previous games, one can show that all users follow a threshold strategy at equilibrium. However, in this game, the users do not only differ by their lead times, but also by the number of available servers they observe. Thus, in this case, the threshold strategy function maps two variables, the lead time t and the number of available servers n, to a decision AR or AR'. For each value of n, we have a different threshold τ_n. More formally:

$$\sigma(t, n) = \begin{cases} AR & \text{if } t > \tau_n, \\ AR' & \text{if } t \leq \tau_n. \end{cases} \tag{7}$$

Users are more likely to get service when observing more available servers. Hence, one can prove that

$$0 \leq \tau_{n-1} \leq \tau_n \leq 1, \quad \forall n \in \{2, N\}. \tag{8}$$

That is, if a user is better off making AR when observing n available servers, she is also better off making AR when observing $n - 1$ available servers.

In order to find the equilibrium structure, we define N virtual users. The n-th threshold user is a virtual user with lead time equals to τ_n that observes n available servers. Assume that all users follow the same threshold strategy function. In this case, the probability that the n-th threshold user gets service, if she does not make AR, depends only on the threshold τ_n. We denote this probability by $\pi_n(\tau_n)$ (see [10] for an explicit formula). Showing that $\pi_n(\tau_n)$ is a non-increasing function of τ_n is sufficient to prove that, unlike the previous games, this game has a unique equilibrium.

To find the equilibrium, one should first check whether $\pi_N(1)$ is greater or smaller than $1 - C$. If it is greater, then all users that observe N available servers are better off not making AR. Thus, the system keeps staying with N available servers and *none-make-AR* is the unique equilibrium. Otherwise, the unique equilibrium is *some-make-AR*. In a *some-make-AR* equilibrium, each threshold user is indifferent between the actions AR and AR'. Thus, the N thresholds are the unique solution of the set of N equations:

$$1 - C = \pi_n(\tau_n), \quad \forall n \in \{1, ..., N\}. \qquad (9)$$

To illustrate the equilibrium, consider a game with $N = 6$ servers, a Poisson distributed demand with mean $\lambda = 6$ and an AR cost $C = 0.15$. To find the equilibrium strategy, we first check which type of equilibrium the game satisfies. Since $C < 1 - \pi_6(1) = 0.16$, we determine that the game has a *some-make-AR* equilibrium. We then solve Equation (9) and get that the unique solution is $\{\tau_1 = 0.056, \tau_2 = 0.213, \tau_3 = 0.395, \tau_4 = 0.584, \tau_5 = 0.777, \tau_6 = 0.973\}$.

Due to the complexity of the equilibrium structure (each equilibrium consists of N thresholds instead of one), finding a closed-form expression for the expected number of reservations is challenging. Thus, in order to determine which information sharing policy maximizes the number of reservations, we resort to simulations.

We consider two systems, both with $N = 10$ servers and Poisson distributed demand. In the first system, $\lambda = 10$, while in the second system $\lambda = 12$. In each system, we consider AR costs between 0.01 and the highest cost that yields a *some-make-AR* equilibrium. For each combination of mean demand and cost, we derive the equilibrium strategy function.

We then run 10,000 simulations for each combination using the corresponding equilibrium strategy function. Figure 4 shows that, for each combination of AR cost and mean demand, the average number of reservations is higher when no information is shared.

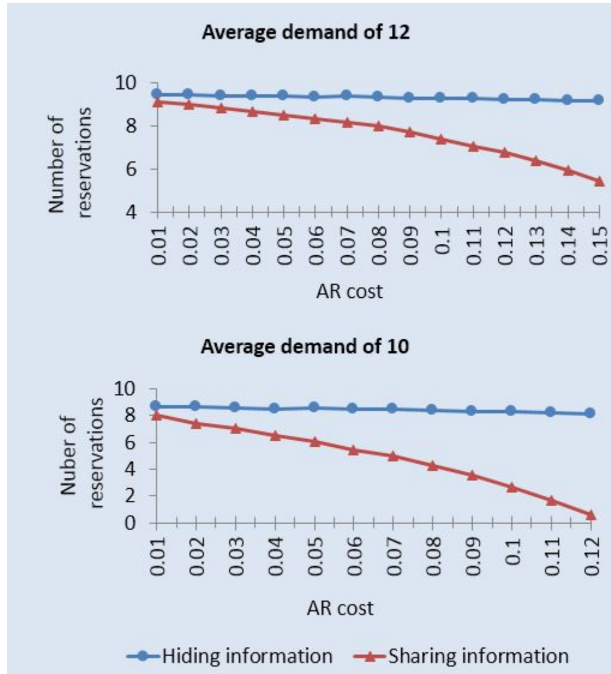

Figure 4: The number of reservations as a function of the AR cost in a system with $N = 10$ servers. Hiding information about the number of servers available brings more users to make advance reservation, on average.

The simulation results also indicate that the gap between the outcomes of the two different information sharing policies increases with the AR cost. When the cost is low, the motivation to make reservations is high and almost all servers are reserved, regardless of the policy. As the reservation cost increases, the gap becomes more significant. For example, consider an AR cost $C = 0.12$ and an average demand $\lambda = 10$. In this case, the average number of reservations when the information is not shared is more than 10 times higher than when the information is shared.

Impact of Learning

The AR games analyzed so far assume that all users follow an equilibrium strategy. To relax this assumption, we study a dynamic version of the game, in which users initially follow an arbitrary threshold strategy. Our goal is to find whether the system converges to an equilibrium or cycles. If it converges and multiple equilibria exist, we wish to find to which equilibrium the system converges.

In dynamic games, the game repeats many times. At each iteration, players update their strategies, after observing the actions of players at previous iterations. Different learning models differ by the types of historical data players obtain throughout the game. They also differ by the way the players estimate the strategies that will be followed by the rest of the players.

We next focus on *best response dynamics*. In this type of learning, the players observe the actions made by other players in the previous iteration and assume that the other players will not change their strategy in the current iteration.

Most of the literature assumes that the same set of players participate at each iteration. However, in the framework of AR games, this assumption does not hold. Thus, we assume that a new set of players participate in each iteration. At each iteration, users are informed about the fraction of users that did not make AR in the previous iteration. Users exploit this information to estimate the threshold strategy followed in the previous iteration. Each user then chooses an action that maximizes her own expected payoff, assuming that she is the only one deviating from the estimated threshold strategy.

One can show that the best response of each user to an arbitrary threshold strategy is also a threshold strategy. Thus, we can define a joint best response function that describes the actions of all users to a given threshold estimation. We denote the threshold followed at iteration i by α_i and its estimator by $\hat{\alpha}_i$. We denote the joint best response function by $BR(\cdot)$. The dynamic process is then given by:

$$\alpha_i = BR(\hat{\alpha}_{i-1}). \tag{10}$$

By studying this dynamic process, one can show that [11]:

- If the game has a unique *some-make-AR* equilibrium with threshold τ, then the game does not converge to an equilibrium. Instead, it cycles in the range $[0, \tau]$.
- If the game has multiple equilibria including a *none-make-AR* equilibrium, then it is guaranteed to eventually converge to the *none-make-AR* equilibrium.

From the result above, we deduce that if the AR cost corresponds to a fee charged by the provider, then the provider is better off being risk-averse (i.e, to charge a low fee with guaranteed revenue).

Figure 5 illustrates the convergence process in a system with $N = 10$ servers, a Poisson distributed demand with mean $\lambda = 10$ and reservation cost $C = 0.215$. The system has three equilibria: $\tau = 0.29$, $\tau = 0.53$ and $\tau = 1$. Starting with $\hat{\alpha}_i = 0$, (i.e., initially all users make AR), the dynamic threshold $\hat{\alpha}_i$ cycles for some period of time, but eventually converges to the equilibrium $\tau = 1$ (i.e., none of the users makes AR). If $\tau = 1$ were not an equilibrium, then the system would keep cycling and yielding positive revenue for the provider.

Conclusions and Future Work

In this article, we propose a game-theoretic framework, called AR games, to study the behavior of users in systems that support advance reservations. Using this framework, we obtain several insights that can serve as guidelines in the design of AR systems.

In particular, we show that all Nash equilibria are of threshold form. Only users with lead time greater than the threshold make AR. Hence, two types of equilibrium prevail: *some-make-AR* and *none-make-AR*.

The analysis of dynamic AR games shows that a provider aiming at maximizing his revenue is likely to end up with no revenue, if a *none-make-AR* equilibrium exists for the same AR fee. Thus, a provider should rather opt being risk-averse in that case.

We also show that it is in the best interest of the provider to charge AR fees only from users granted services, and not to inform users about the number of servers available.

This work opens several areas for future research. For instance, one open question is how to incorporate

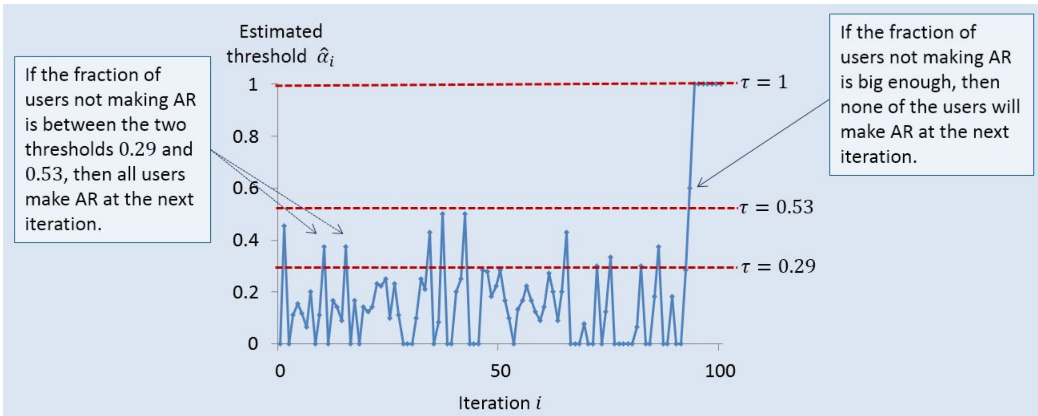

Figure 5: An illustration of a dynamic game [11]. If the game has a *none-make-AR* equilibrium, then eventually, the strategy of the users converges to it.

re-trials by users not granted service. The analysis of a system with re-trials is complex due to the dependencies of the demand across different time slots. The model could also be generalized to allow users to reserve multiple servers in one slot or to reserve servers across multiple slots. Another challenge is to find dynamic pricing schemes that would increase the revenue of the provider, especially when users have information about server availability. The framework of AR games should prove useful in exploring all these interesting problems.

References

[1] C. Guok *et al.*, "Intra and interdomain circuit provisioning using the oscars reservation system," in *Broadband Communications, Networks and Systems, 3rd International Conference on*, IEEE, 2006, pp. 1–8.

[2] N. Charbonneau and V. M. Vokkarane, "A survey of advance reservation routing and wavelength assignment in wavelength-routed WDM networks," *Communications Surveys & Tutorials, IEEE*, vol. 14, no. 4, 2012, pp. 1037–1064.

[3] B. Sotomayor, *et al.*, "Virtual infrastructure management in private and hybrid clouds," *Internet computing, IEEE*, vol. 13, no. 5, 2009, pp. 14–22.

[4] D. Xie, *et al.*, "The only constant is change: incorporating time-varying network reservations in data centers," *ACM SIGCOMM Computer Communication Review*, vol. 42, no. 4, 2012, pp. 199–210.

[5] D. Niu, C. Feng, and B. Li, "Pricing cloud bandwidth reservations under demand uncertainty," in *ACM SIGMETRICS Performance Evaluation Review*, vol. 40, no. 1, 2012, pp. 151–162.

[6] R. A. Guérin and A. Orda, "Networks with advance reservations: The routing perspective," in *Computer and Communications Societies (INFOCOM), IEEE*, vol. 1, 2000, pp. 118–127.

[7] R. Cohen, N. Fazlollahi, and D. Starobinski, "Path switching and grading algorithms for advance channel reservation architectures," *Networking, IEEE/ACM Transactions on*, vol. 17, no. 5, 2009, pp. 1684–1695.

[8] N. Fazlollahi and D. Starobinski, "Distance vector-based advance reservation with delay performance guarantees," *Theory of Computing Systems*, 2015, pp. 1-28.

[9] E. Simhon and D. Starobinski, “Game-theoretic analysis of advance reservation services,” in *Information Sciences and Systems (CISS), 48th Annual Conference on*, IEEE, 2014, pp. 1-6.

[10] E. Simhon and D. Starobinski, “On the impact of sharing information in advance reservation systems,” in *The 10th Workshop on the Economics of Networks, Systems and Computation (NetEcon)*, ACM, 2015, pp. 82-82.

[11] E. Simhon,*et al.*, “Pricing in dynamic advance reservation games,” in *Fourth IEEE Workshop on Smart Data Pricing (SDP)*, 2015, pp. 546-551.

[12] M. I. Reiman and Q. Wang, “An asymptotically optimal policy for a quantity-based network revenue management problem,” *Mathematics of Operations Research*, vol. 33, no. 2, 2008, pp. 257–282.

[13] D. Bertsimas and R. Shioda, “Restaurant revenue management,” *Operations Research*, vol. 51, no. 3, 2003, pp. 472–486.

[14] J. Nasiry and I. Popescu, “Advance selling when consumers regret,” *Management Science*, vol. 58, no. 6, 2012, pp. 1160–1177.

[15] R. Hassin and M. Haviv, *To queue or not to queue: Equilibrium behavior in queueing systems*, Springer Science & Business Media, 2003.

Biographies

Eran Simhon received his B.Sc and M.Sc in Industrial Engineering and Management from the Technion - Israel Institute of Technology in 2006 and 2012, respectively. He has four years of work experience as a supply chain engineer in the semiconductor industry. Currently he is a Ph.D. candidate in the Division of Systems Engineering at Boston University. His research interests include queueing theory, game theory and network economies.

David Starobinski is a Professor of Electrical and Computer Engineering at Boston University, with a joint appointment in the Division of Systems Engineering, and a Faculty Fellow at the U.S. DoT Volpe National Transportation Systems Center. He received his Ph.D. in Electrical Engineering from the Technion - Israel Institute of Technology, in 1999. In 1999–2000, he was a visiting post-doctoral researcher in the EECS department at UC Berkeley. In 2007–2008, he was an invited Professor at EPFL (Switzerland). Dr. Starobinski received a CAREER award from the U.S. National Science Foundation (2002), an Early Career Principal Investigator (ECPI) award from the U.S. Department of Energy (2004), the best paper award at the WiOpt 2010 conference, and the 2010 BU ECE Faculty Teaching Award. He was an Associate Editor of the IEEE/ACM Transactions on Networking from 2009 to 2013. His research interests are in wireless networking, network economics, and cybersecurity