The Evolution of Cooperation in File Sharing P2P Systems: First Steps

María Esther Sosa-Rodríguez and Elizabeth Pérez-Cortés

Universidad Autónoma Metropolitana Unidad Iztapalapa
Electrical Engineering Department

Abstract. In order to cope with the free-riding problem in file sharing P2P systems, two kinds of incentive mechanisms have been proposed: reciprocity based and currency based. The main goal of this work was to study the impact of those incentive mechanisms in the emergence of cooperation in file sharing P2P systems. For each kind of incentive mechanism we designed a game and the outcome of this game was used as a fitness function to carry out an evolutionary process. We were able to observe that the Currency Game obtains an enough cooperative population slightly faster than the Reciprocity Game but, in the long run, the Reciprocity Game outperforms the Currency Game because the final populations under the former are consistently more cooperative than the final populations produced by the latter.

Keywords: P2P systems, free-riding, incentive mechanisms, evolutionary algorithm.

1 Introduction

Peer to peer systems (P2P) are composed by autonomous nodes that organize themselves in order to share resources. In P2P systems, each node is a potential server and a client and this characteristic makes them naturally scalable and fault tolerant [1][2]: more peers means more resources and a higher service capacity meanwhile, if a peer leaves the system or fails, another one is able to provide the service. The P2P model is only challenged by peers autonomy that entitles them to decide when to join and leave the system and also what resources they share during their participation. In other terms, scalability and fault tolerance of P2P applications largely depend on the level of cooperation of autonomous nodes.

We place our research in the context of a file sharing P2P (FSP2P) system where files are uploaded by owners and cooperative peers store copies in order to be, later, able to serve download requests. The value of a FSP2P is associated with the amount of shared content[1]. The more content is in the system, the better it is for the clients.

As FSP2P applications became popular, peers having rational behavior appeared. Those peers are named *free-riders* because they attempt to download

[1] In this work, we use indistinctly the terms content and file as well as node and peer.

I. Batyrshin and M. González Mendoza (Eds.): MICAI 2012, Part II, LNAI 7630, pp. 151–162, 2013.

files at the lowest cost by deviating from standard protocols [3]. Free-riders pretty soon drove the attention of the community to the design of incentive mechanisms to avoid them. An incentive mechanism mainly encourages nodes to cooperate, detects free-riders and prevents them to get services.

Roughly speaking, there are two types of incentive mechanisms consistently identified in the literature [3][4]: reciprocity based and currency based. In reciprocity based incentive mechanisms, peers give to receive. Each peer has information about the behavior of other peers and uses this information to decide whether to provide or not a service to them. Reciprocity can be direct or indirect. It is direct if peers only consider each other behavior during the current interaction, as in the case of BitTorrent [5]. In the indirect case, information on the past behavior of each peer is collected and used in the evaluation. In currency based incentive mechanisms, peers pay for the service and are paid for their contributions with a currency that can be virtual and/or real.

The main goal of this work was to study, under an evolutionary approach, the impact of incentive mechanisms in a FSP2P system. We modeled the interaction of nodes as a game [6] where nodes follow predefined strategies, issue download requests and an incentive mechanism is applied. The outcome of this game is the successful download ratio and this measure is used as fitness function to carry out an evolutionary process. We studied the impact of each incentive mechanism in the degree of cooperation in the system from two aspects, the way it augments the collective capacity of the system to store all the files and how it modifies the initial composition of the population.

The rest of this article is structured as follows, in section 2 we describe the previous efforts to model P2P systems using games. In section 3, we present the Reciprocity Game and the Currency Game that model the FSP2P system for the two classes of incentive mechanisms. The evolutionary process is presented in section 4 and the experiments we carried out and their results are described and discussed in section 5. Finally, conclusions are outlined in section 6.

2 Related Work

In a FSP2P system, the main decision a peer faces, to store a copy of a file or not, is a strategical situation. Peers have to chose between their individual interests and those of the community: if a peer stores a copy to make it available to the others, it pays some costs but contributes to the value of the system, if it does not, it has no costs but the system may lose value. This situation can be modeled as a game as defined in Game Theory [6]: peers are the players, that have two possible actions and obtain a payoff that depends on the choices made by other peers and the own choice.

During the last decade, several research efforts [7][8][9] have used the Game Theory framework of knowledge to model the strategical situation that peers face. This approach considers that peers are all rational players that always seek the action that provides the best outcome for themselves. The goal of this approach is to find a set of actions where each peer plays the best possible response, i.e. a Nash equilibrium, and use this result to predict the behavior of the system.

Although several assessment studies have confirmed the presence of rational peers in P2P systems, the existence of non rational behaviors, as like altruism has also been observed [10][11]. This reality can be better modeled using Evolutionary Game Theory (EGT) as proposed in [12][13][14]. EGT eliminates the hypothesis of rationality and includes the dynamic aspect of P2P systems. In these works, peers are players following an encoded strategy whose survival depends, on the one hand, on the utility obtained by the strategies when interacting with each other and, on the other hand, on the proportion of the population that follows each strategy. EGT allows us to predict the prevalence of strategies throughout time and the stability of the configuration of a population.

In our case, we are interested in the emergence of cooperative behavior as a result of the application of an incentive mechanism; we do not propose specific strategies but we do want to study the resulting ones. In consequence, we work with a EGT-like approach in the sense that we propose a game for each incentive mechanism where rationality is not assumed but players follow an inherited strategy, and then, the utility that each peer obtains is used as a fitness function for an evolutionary process.

3 The Cooperation Games

In this section, after defining some important concepts and the elements of every game, we present a Generic Cooperation Game (GCG) that models the FSP2P system independently of the incentive mechanism. Then, we propose a game for each class of incentive mechanisms, called Reciprocity Game and Currency Game. They are both based in the GCG and their differences, born from the incentive mechanism they use, are presented in the subsections 3.4 and 3.5 respectively.

3.1 Preliminary Concepts

Availability of peers and files are central concepts in FSP2P systems. On the one hand, the *availability of a peer* models its transient character and it is a number that tell us how likely is to find the peer on the FSP2P system in a given instant. This number depends on the owner and it has been shown that its value follows a Weibull distribution [15]. On the other hand, the *availability of a file* is also a probability, the one of finding the file in a given instant and, in this case, it is determined by the replication technique and the availability of the peers that participate in the storage. In this first work, we considered full replication, that is, when a peer decides to cooperate, it stores and distributes a full copy of the file. In this case, the availability of the file f^i is given by the next expression:

$$f^i_{\text{Availability}} = 1 - \prod_{j=1}^{k} (1 - p^j_{\text{Availability}}) \tag{1}$$

where k is the number of current copies and $p^j_{\text{Availability}}$ is the availability of the peer that stores the copy number j.

We consider that each shared file has availability requirements and if the system is not able to provide them, the file is lost for practical uses. Also we have to consider that the actual amount of copies of a file that a FSP2P system stores depends, not only on the amount of peers willing to store and share it, but also on other factors like performance, cost, etc. In other words, meanwhile a peer can chose to store a file and share a copy or not, this does not mean that the file is stored in that peer or even stored in the system. Therefore, an *storage event* is recalled by a peer as a triplet: $\langle Coopperate, SystemAvailable, PeerAvailable\rangle$ where $Cooperate$ is T if the peer determined to store and share a copy of the file and F otherwise, $SystemAvailable$ is T when the file is not lost and F when it is and finally, $PeerAvailable$ is T when the peer stores the file and F otherwise. This last entry is meaningful only when the second one is T. Lets notice that an storage event can be represented by a three bit string. A peer is able to keep track of the recent history in a *memory* and might use this information to select its next action.

Clearly, when a peer participates in a FSP2P system, it intends to download content. Therefore, it will issue download requests and some of them will succeed meanwhile others will not. The later situation arises when the file was lost or when the FSP2P system decides that this peer has not been cooperative enough to deserve the service. Each peer has a $successful\ download\ ratio$ (sdr) that is calculated as the quotient between the successful downloads and the total amount of issued download requests. The *sdr* value represents how fit is the strategy of the peer for the particular FSP2P system. We said that a peer is more successful than another peer if it has a bigger *sdr* value.

Finally we define the *success* of a FSP2P system as the percentage of files that are stored in the system w.r.t. those that were submitted to it. A FSP2P system is *cooperative enough* when it is able to store the total amount of files 100%. We said that a population X is more cooperative or more successful than another population Z if X is able to keep a bigger percentage of files than Z.

3.2 Players and Strategies

In the GCG, each peer is a player with a *memory* of size m where it tracks down the last m *storage events* and a *strategy* that specifies an action for each possible value of the *memory.* Since a *storage event* is a three bit string (c.f. subsection 3.1), to recall m of them we need $3 * m$ bits.

Formally we define a *strategy* S as a sequence $s_1 \cdots s_{2^{3*m}}$ where $\forall i, s_i \in$ $\{$*share*, *no-share*$\}$. A *share* action indicates that the peer will store and share a copy of a content whereas *no-share* means that it will not. We say that a strategy A is more cooperative than a strategy B if A includes the action *share* more times than B.

Lets notice that the strategy of a *free-rider* is a sequence of *no-share* actions meanwhile the one corresponding to an *altruist* is a sequence of *share* actions. A randomly generated strategy is called a *honest* strategy as, in average, it tends to contain the same amount of *share* and *no-share* actions.

Summarizing, each player has the following attributes:

- *availability*: probability of being in the system at a given instant,
- *requests*: total number of download requested,
- *downloads*: number of successfully downloaded files,
- *sdr*: successful download ratio = $downloads \div requests$,
- *memory*: the binary description of the last m *storage events* and
- *strategy*: sequence of 2^{m*3} actions

Peers have an unlimited storage space so that a cooperative behavior can not be blurred by its storage capacity. We believe that the current size of hard disks allows us to make this assumption without losing generality.

3.3 The Generic Cooperation Game

A GCG is composed by a sequence of N turns. As in [8][9], on each turn an uniquely identified file f^i is submitted to the system to be stored with a $f^i_{\text{RequiredAvailability}}$. A turn has two phases:

1. Storage phase: during this phase, each peer determines its action independently and simultaneously using its *memory* and its *strategy*. Depending on the result of this phase and the incentive mechanism, the file is stored in the system or lost.
2. Download phase: during this phase, each peer randomly decides if it issues or not a request to download a file f^r where $r \in \{1...i-1\}$. If it does, its *requests* counter is augmented by one. It is up to the incentive mechanism to decide whether a download request is granted or not. When the request is granted and the file is stored in the system, the peer increments its *downloads* counter by one.

At the end of the game, after the N turns, we calculate the *sdr* of each peer (the fitness of its strategy) and the *success* of the FSP2P system.

3.4 The Reciprocity Game

In reciprocity based incentive mechanisms, each peer has a *reputation* that is built during the game. In our model, the *reputation* is the proportion of files a peer is willing to store even if it is not elected to do so.

In the Reciprocity Game (RG) a peer will select the action indicated in *strategy*[*memory*]. Once all the peers have chosen their actions, using the expression (1), we calculate the potential availability of the file $f^i_{\text{PotentialAvailability}}$ considering that we store a copy on each cooperative peer, then we compare this number against the $f^i_{\text{RequiredAvailability}}$. If

1. $f^i_{\text{RequiredAvailability}} = f^i_{\text{PotentialAvailability}}$, the file is stored on the system.
2. $f^i_{\text{RequiredAvailability}} < f^i_{\text{PotentialAvailability}}$ a subset of the cooperative peers, large enough to ensure the requested availability, are randomly elected to store the file.
3. $f^i_{\text{RequiredAvailability}} > f^i_{\text{PotentialAvailability}}$ the file is lost.

In the RG a request from the peer will be granted if it has an enough good reputation. The amount of requests that are granted for a peer is in direct proportion to its reputation, a peer having a reputation between $0.2 * i$ and $0.2 * (i + 1)$ included, for $i \in \{0, 1, 2, 3, 4\}$, will be granted up to $20 * (i + 1)\%$ of its download requests.

3.5 The Currency Game

In currency based incentive mechanisms, peers pay for the service and are paid for their contributions with a virtual currency.

In this scenario, the Currency Game (CG), peers have two additional attributes, a *storage price* and a *capital.* The first one represents the amount of currency the owner of a file will be charged to store a copy in the peer. This amount is related to the availability of the peer. The *capital* represents the current amount of currency the peer owns. As for files, they have two additional attributes called *rent* and *download price* that represent respectively, the amount of currency they can pay to be stored and the price they will demand to be downloaded.

A peer will cooperate if the action indicated in *strategy*[*memory*] is *share* and the *rent* of the file is at least equal to the *storage price* of the peer. Cooperative peers will be ordered on the base of its *availability* per *storage price* ratio and then, the best of them will be chosen until the *availability* of the file is reached or the *rent* is exceeded. In the former case the elected peers are paid *storage price.* In the latter case, the file is lost and nobody is paid.

A request of a peer will be granted if it has enough currency to pay the *download price* of the file.

4 Evolving Cooperative Behavior

Up to now we have defined two games, the RG and the CG, that provide a way to evaluate the fitness of strategies in a FSP2P system. This section is devoted to the description of the elements of the evolutionary process that we applied in order to obtain more fitted strategies, that is, strategies that allow peers to maximize the amount of downloaded files under a specific incentive mechanism.

The elements of the evolutionary process presented here are largely inspired in the pioneer work of R. Axelrod [16] on the evolution of cooperation in the Prisoner's Dilemma game. For some of them we tested different techniques[2] but did not obtain better results.

[2] Due to the lack of space these tests are not presented in this paper.

Evolutionary Process

Given a number of generations, a game G, a number of rounds N, a size for the population T and a specific proportion of types of strategies in it (c.f. 3.2), we perform the following steps:

1. Initialization: we build a population of nodes whose strategies are compliant with the given specification. Those strategies are the individuals of the evolutionary process. The initial population becomes the current population.
2. Evaluation: to evaluate individuals, the current population plays the game G during N rounds and, at the end, we calculate the *success* of the population and the *sdr* (successful download ratio) of each node ($downloads \div requests$).
3. Ranking: nodes are separated in three groups depending on its *sdr* value as follows (σ stands for the standard deviation of the *sdr* values):
   i) if $sdr < (average(sdr) - \sigma)$ they go to group 0,
   ii) if $(average(sdr) - \sigma) \leq sdr \leq (average(sdr) + \sigma)$ they go to group 1 and
   iii) if $sdr > (average(sdr) + \sigma)$ they go to group 2.
4. Selection: $T/2$ couples of parent nodes are selected. If a peer is in group 1 it is selected once and if it is in group 2 it is selected twice. Peers in group 0 are discarded.
5. Crossover and Mutation: each couple of parents produces a couple of children nodes. In order to obtain the children, two new nodes are created and their strategies are obtained by the application, with a probability P_c, of one point crossover on the strategies of the parents and then, performing on the resulting strategies an uniform mutation with a probability P_m. The *availability* (and the *storage price* for the CG) of the children nodes are inherited from the more successful parent.
6. The whole set of children becomes the current population and the process is repeated since the Evaluation step.

The evolutionary process stops when the prefixed number of generations is reached.

5 Experiments and Results

In order to study the impact of the incentive mechanisms, we implemented a modular Game Simulator using Java 1.6 and we executed the evolutionary process on predefined populations. For the sake of comparison, besides the CG and the RG, we also programmed a game without any incentive mechanism, named Free Game (FG) where download requests are granted randomly.

5.1 Parameters

All the experiments were carried out for a population of 500 nodes with a *memory* of size 3 and strategies of size 512. The *availability* of each node was obtained using a Weibull distribution with form parameter equal to 1. In the case of the Currency Game, initial *capital* was 0 and the *storage price* was calculated randomly on a range that depends on the *availability* of the node as follows: for a node whose *availability* is in $[1\%, 34\%)$, the range was $[0.01, 0.34)$, if the *availability* is in $[34\%, 67\%)$ then, the *storage price* is in $[0.34, 0.67)$ and if the *availability* is in $[67\%, 100\%]$, the *storage price* is in $[0.67, 1.0]$. In the case of the Reciprocity Game, the initial *reputation* was 0.

Each game was played for 1000 turns for files with a required availability uniformly distributed in the range $[1\%, 100\%]$. In the case of the Currency Game, the *rent* was uniformly distributed in the range $[0.1, 10]$ and the *download price* in $[0.01, 1]$.

Populations underwent evolutionary process during 500 generations using a crossover probability $P_c = 0.25$ and a mutation probability $P_m = 0.001$.

Each experiment was repeated thirty times and the value that is considered in the plots is the average value on the thirty repetitions.

5.2 Impact on a Free-Riders Population

The goal of this first experiment was to observe the impact of incentive mechanisms on a population of free-riders. First, we wanted to observe how fast each mechanism drives the population to a state where it is able to keep the 100% of the submitted files and second, how the fitness, the successful download ratio of the strategies, evolves under each mechanism.

In Figure 1 we plot the amount of stored files in each generation for the different games. We can observe a clear difference of the speed of the evolution of cooperation between the populations using incentive mechanisms (RG and CG) and the one where none is used (FG). For this configuration, CG shows a faster, although slightly, evolution in the success of the population and a better stability.

In Figure 2 we can observe the changes in the average successful download ratio of peers under the evolutionary process for each game. In this case, the best growing rate is the one of RG that grows steadily until it reaches the perfect value; the second best is for FG that stabilizes around 0.5, this result is coherent with the fact that, in this game, requests are granted on an uniformly distributed random base. Finally, we can observe that, for CG, the average *sdr* never grows, in fact, as there is a fixed amount of currency being distributed among the peers in the system, no matter how it is distributed, the average will not change. This raises an important issue for the configuration of currency based incentive mechanisms because, if there is too much competition to obtain the currency, as in our experiment, peers having a poor successful download ratio will tend to leave the FSP2P system.

5.3 Impact on the Composition of the Final Population

The goal of this experiment was to observe how different initial populations are changed by the evolutionary process under the RG and the CG. The considered

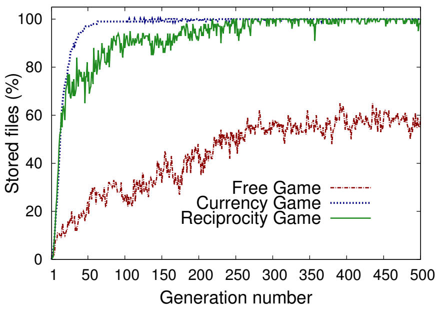

Fig. 1. Success of the population

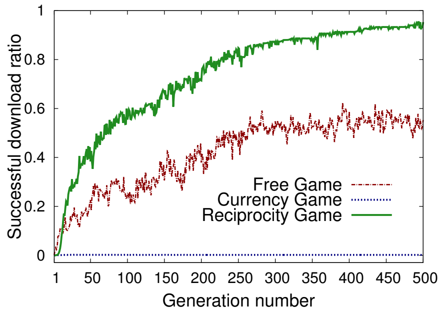

Fig. 2. Successful download ratio

populations, named $P1, P2, P3$ and $P4$ are as follows: $P1$ is a free-rider population, $P2$ is composed by 98% of free-riders and 2% of honest peers, $P3$ is 98% of free-riders and 2% of altruist peers and $P4$ is a population of honest peers. P2 and P3 allowed us to study how a small proportion of honest or altruist peers could boost the emergence of cooperation and drive a population to behave like a honest population (P4).

Programmed Behavior

After 500 generations, we classified the peers in the final population depending on the percentage of *share* actions included in its strategy. We made five groups, each one corresponding to a 20% partition. The results are shown in Figures 3 and 4 for the CG and the RG respectively.

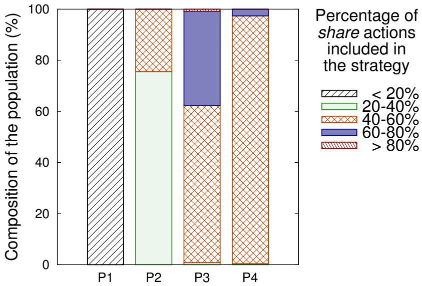

Fig. 3. Programmed behavior CG

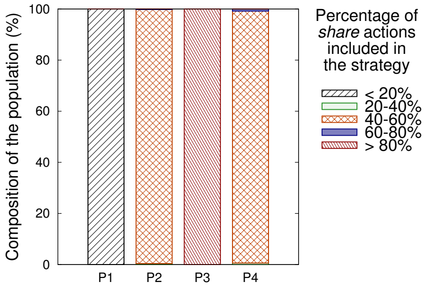

Fig. 4. Programmed behavior RG

For P1, the final population was totally composed of strategies with at most 20% of *share* actions under both incentive mechanism. For P2, in the final population for CG we observe a majority of peers having strategies with at most

40% of *share* actions meanwhile, in the case of RG, it is totally composed by peers with strategies with a percentage of *share* actions between 40% and 60%. For P3, under CG the final population is mainly composed of strategies having more than 40% and less than 80% *share* actions whereas, under RG, it is fully composed of strategies with more than 80% of *share* actions. Finally, for the honest peer population, P4, we do not observe meaningful differences in the composition of the final population. Under both mechanisms the majority of the peers have strategies with a percentage of *share* actions between 40% and 60%.

Summing up, the strategies of the populations produced by RG are more cooperative, include the action *share* more times, than those produced by CG.

Exhibited Behavior

As explained before, in our models a peer determines its actions using the current value of its memory and its strategy, but, even when its strategy is composed of a certain proportion of *share* and *no-share* actions, the peer can behave, for example, as a free-rider (playing always *no-share*) or as an altruist (play always *share*). In other words, peers can exhibit a behavior that differs strongly from the one that appears in its strategy. In consequence, we decide to observe the exhibited behavior, in order to do so, we classified the peers of the final population depending on the percentage of *share* actions they chose to play in the last game. Again, we have a partition at every 20%. The results are shown in Figures 5 and 6 for the CG and the RG respectively.

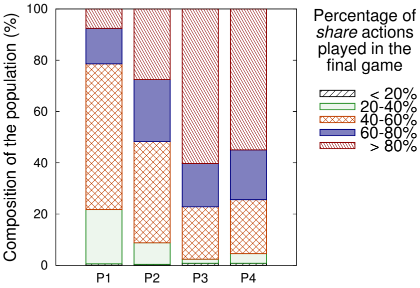

Fig. 5. Exhibited behavior in CG

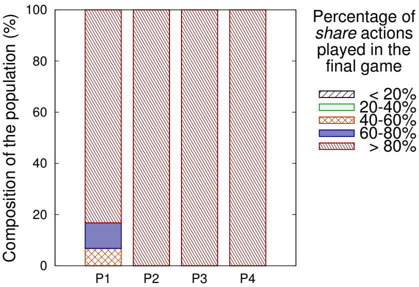

Fig. 6. Exhibited behavior in RG

We can observe that exhibited behaviors include more cooperation, in both RG and CG, than programmed behaviors. On the one hand, under CG, P1 behavior ends up far away from the programmed behavior with more than the 99% playing the action *share* at least 20% of the times and, in the case of P2, P3 and P4, more than 85% chose the action *share* at least 40% of the times. On the other hand, under RG, P1 has only a small fraction of peers that played the action *share* less than 80% whereas in P2, P3 and P4, peers selected the action *share* at least 80% of the times.

Summarizing, independently of the initial composition of the population, the peers of the final populations evolved by RG behave more cooperatively than the peers in the final populations obtained using CG.

Impact of Honesty and Altruism

We can observe, in Figures 4 and 6, that under RG it is enough to introduce 2% of honest peers (P2) to obtain a final population that is close, in its programmed and exhibited behavior, to a honest population (P4). In the case of CG, see Figures 3 and 5, it is the introduction of altruist peers (P3) what produces a composition closer to P4.

6 Conclusions

In this article we studied the evolution of cooperation in a FSP2P system using two different incentive mechanisms. For each one of them we designed a game where peers play inherited strategies and are able to successfully download a portion of the files that they required. Using this measure as a fitness function, we applied an evolutionary process and observed the effect of the incentive mechanisms in the emergence of cooperation, the fitness function and the changes on the composition of the population.

Our results clearly showed that using an incentive mechanism encourage the cooperative behavior. In the case of our proposals, the Currency Game evolves, slightly faster than the Reciprocity Game, an enough cooperative population. However, if the parameters of the system involving currency are not well tuned, this mechanism could lead to the abandon of the FSP2P system due to the steady low successful download ratio of peers. This situation does not arise in the case of the Reciprocity Game.

Also, we were able to observe that, for four specific types of populations, in the quest of cooperative behavior, the Reciprocity Game outperforms the Currency Game. In fact, the final populations under the Reciprocity Game are consistently more cooperative than final populations of the Currency Game. This conclusion is supported by two facts: i) the *share* action appears more times in the strategies of peers in the final populations of the Reciprocity Game than in those of the Currency Game and, ii) in the final Reciprocity Game the action *share* was played a bigger percentage of times than in the final Currency Game, no matter how the initial population was composed.

In the future, we intend to perform a full parametric study on the models of the games and on the evolutionary process, as well as to test different fitness functions. The final goal is to use this approach to study incentive mechanisms used in real systems and take advantage of the results to enhance them.

Acknowledgments. The first author acknowledges support from CONACyT to pursue graduate studies in Science and Technologies of the Information at UAM-Iztapalapa.

References

1. Androutsellis-Theotokis, S., Spinellis, D.: A Survey of Peer-to-Peer Content Distribution Technologies. ACM Computing Surveys **36**(4) (December 2004) 335–371
2. Passarella, A.: Review: A Survey on Content-Centric Technologies for the Current Internet: CDN and P2P Solutions. Computer Communications **35**(1) (January 2012) 1–32
3. Zhang, K., Antonopoulos, N., Mahmood, Z.: A Review of Incentive Mechanism in Peer-to-Peer Systems. In: Proceedings of the First International Conference on Advances in P2P Systems, 2009. AP2PS '09, IEEE (October 2009) 45–50
4. Feldman, M., Chuang, J.: Overcoming Free-riding Behavior in Peer-to-Peer Systems. SIGecom Exchanges **5**(4) (July 2005) 41–50
5. Cohen, B.: Incentives Build Robustness in BitTorrent. In: Proceedings of the First International Workshop on Economics of Peer-to-Peer Systems. (June 2003)
6. Peters, H.: An Introduction to Evolutionary Games. Springer, Berlin Heidelberg (2008)
7. Geels, D., Kubiatowicz, J.: Replica Management Should Be a Game. In: Proceedings of the 10th Workshop on ACM SIGOPS European Workshop. EW 10, ACM (2002) 235–238
8. Chun, B.G., Chaudhuri, K., Wee, H., Barreno, M., Papadimitriou, C.H., Kubiatowicz, J.: Selfish Caching in Distributed Systems: A Game-Theoretic Analysis. In: Proceedings of the Twenty-Third Annual ACM Symposium on Principles of Distributed Computing. PODC '04, ACM (2004) 21–30
9. Khan, S.U., Ahmad, I.: A Pure Nash Equilibrium Guaranteeing Game Theoretical Replica Allocation Method for Reducing Web Access Time. In: Proceedings of the 12th International Conference on Parallel and Distributed Systems - Volume 1. ICPADS '06, IEEE Computer Society (2006) 169–176
10. Kaune, S., Tyson, G., Pussep, K., Mauthe, A., Steinmetz, R.: The Seeder Promotion Problem: Measurements, Analysis and Solution Space. In: Proceedings of 19th International Conference on Computer Communications and Networks 2010. ICCCN '10, IEEE (August 2010) 1–8
11. Hughes, D., Coulson, G., Walkerdine, J.: Free Riding on Gnutella Revisited: The Bell Tolls? IEEE Distributed Systems Online **6** (2005) 1–6
12. Ma, R.T.B., Lee, S.C.M., Lui, J.C.S., Yau, D.K.Y.: Incentive and Service Differentiation in P2P Networks: A Game Theoretic Approach. IEEE/ACM Transactions on Networking **14**(5) (October 2006) 978–991
13. Zhang, Q., Xue, H.F., Kou, X.D.: An Evolutionary Game Model of Resources-Sharing Mechanism in P2P Networks. In: Proceedings of the Workshop on Intelligent Information Technology Application 07. IITA '07, IEEE Computer Society (2007) 282–285
14. Wang, Y., Nakao, A., Vasilakos, A.V., Ma, J.: P2P Soft Security: On Evolutionary Dynamics of P2P Incentive Mechanism. Computer Communications **34**(3) (March 2011) 241–249
15. Stutzbach, D., Rejaie, R.: Understanding Churn in Peer-to-Peer Networks. In: Proceedings of the 6th ACM SIGCOMM Conference on Internet Measurement. IMC '06, ACM (2006) 189–202
16. Axelrod, R.: Evolving New Strategies: The Evolution of Strategies in the Iterated Prisoner's Dilemma. In: Genetic Algorithms and Simulated Annealing, Pitman (1987) 32–41