Validating Peer-to-Peer Storage Audits with Evolutionary Game Theory

Nouha Oualha and Yves Roudier

EURECOM, Sophia Antipolis, France
{nouha.oualha, yves.roudier}@eurecom.fr

Abstract. This paper discusses the efficiency of audits as incentives for reducing free-riding in P2P storage applications, that is, lessening the number of peers that can store their data without contributing to the storage infrastructure. Audits are remote data verifications that can also be used to decide whether to cooperate with a given peer based on its behavior. We demonstrate how an audit-based strategy can dominate or outperform the free-riding strategy by exhibiting the equilibria of our evolutionary game theoretical model of a P2P storage application.

Keywords: evolutionary game, trust establishment, audit, cooperation, P2P storage.

1 Introduction

Peer-to-peer (P2P) is an emerging technology in the storage system area whereby data are distributed in a self-organizing manner to multiple peers instead of using a central storage outsourcing server. Such a distributed storage in particular aims at maintaining a reliable storage without a single point of failure, although without the need for an expensive and energy-consuming storage infrastructure as offered by data centers. Peers volunteer for holding data within their own storage space on a long term basis while they expect a reciprocal behavior from other peers. Just like in P2P file sharing, cooperation incentives have to be used to prevent selfish behaviors whereby peers free-ride the storage system by storing data onto other peers without contributing to the storage infrastructure. Selfishness is however likely more attractive in P2P data storage since unduly gained storage might be used for an extensive period of time contrary to bandwidth which is related to only one file sharing operation. Remote data verification protocols have recently appeared (e.g., [3], [5], [6], [7], [8], [9], [10], and [11]) that aim at auditing peers so as to detect misbehaving ones that claim to hold some data which they in fact destroyed. We claim that such auditing protocols form the basis of an efficient cooperation incentive mechanism. Generally, such a mechanism is proven to be effective if it is demonstrated that any rational peer will always choose to cooperate whenever it interacts with another cooperative peer. We evaluate this mechanism using an evolutionary game model describing the evolution of strategies within large populations as a result of many local interactions, each involving a small number of randomly selected individuals. We propose in this paper an evolutionary game model of a P2P storage system that we use to study under which conditions an audit-based strategy wins over self-interested strategies.

The rest of this paper is organized as follows: in Section 2, an outline of the P2P storage system is first provided. An evolutionary game model of such system is described in Section 3 and the solution of the model, the evolutionary stable strategy, is derived. Results obtained through numerical simulation experiments are finally analyzed in Section 4.

2 A P2P storage system

We consider a P2P storage system in which peers can store their personal data at other peers. The latter, called *holders*, should store data until the *owner* retrieves them. The availability and correctness of stored data is periodically checked by the owner or its delegates, called *verifiers*. However, holders or verifiers may still misbehave in various ways to optimize their own resource usage.

We assume that the storage application enforces a random selection of holders and verifiers (for instance using the method outlined in [13]). This strategy aims at preventing collusions between peers trying to gain an unfair advantage out of the storage system. Behavior assessment relies on an audit-based verification (see [13] for more details) of the cooperation of peers as holders. This assessment is the basis for deciding whether to accept to store or to audit another holder on behalf of a given peer. Audits result from the execution of a remote data possession protocol (see [4]) periodically performed by verifiers. A peer chooses to cooperate with other peers that have proven to be reliable in the past based on audits.

One-stage or repeated games model interactions that only involve the participants to the storage of one piece of data. Modeling the storage of multiple data as a one-stage game would lead to overly complex games otherwise. In contrast, an evolutionary game makes it easy to model the macroscopic aspects of such interactions occurring within populations of peers. An evolutionary game model describes interactions between randomly chosen players, thus practically portraying the random selection of holders and verifiers in the audit-based approach outlined above.

3 Evolutionary game

We propose in this paper an evolutionary game model of a cooperative storage system with which we endeavor to evaluate under which conditions peers using the audit-based strategy will dominate the system.

3.1 Game model

In the proposed system, an owner stores data replicas at r holders. It appoints m verifiers for its data replica that will periodically check storage at holders.

The system is modeled as an evolutionary game. According to Friedman [1], "*an evolutionary game is a dynamic model of strategic interaction with the following characteristics: (a) higher payoff strategies tend over time to displace lower payoff strategies; (b) there is inertia; (c) players do not intentionally influence other players' future actions*".

One-stage game: The one-stage game represents an interaction between one data owner, *r* data holders, and *m* verifiers randomly chosen. Thus, the considered game players are an owner, *r* holders, and *m* verifiers. The one-stage interaction consists of several phases:

- *Storage phase:* the owner stores data at the *r* holders. At this phase, holders may decide to keep data stored or to destroy them depending on their strategy (see next paragraph "Evolutionary game"). Holders that crash or leave the system without any notice are considered as defectors contrary to our previous work [14].
- *Delegation phase:* the owner sends verification information to the *m* verifiers in order to be able to periodically check data at holders. Whether to cooperate with the owner in verifying data is determined by each verifier's strategy (see next paragraph "Evolutionary game").
- *Verification phase:* a verifier can decide whether the holder has been cooperative based on the results of a verification protocol such as [3] and take potential action depending on its strategy. A verifier whose strategy is to cooperate will send the owner the results it obtained by auditing the holder. A non-cooperative verifier may mimic a cooperative strategy by sending a bogus result. Verifiers are not more trusted than other peers and may lie about verification, for instance reporting an absence of response to a challenge for a cooperative holder. A verifier might also be framed by a malicious holder trying to make it appear as a non-cooperative verifier. Some verifiers may also crash or leave the system, and be unable to communicate results of verifications. The owner therefore cannot determine with certainty whether a verifier chose to adopt a cooperative strategy. One negative result from a verifier is also not enough for the owner to decide that the holder is non cooperative. Such a notification may however be used as a warning that the holder may have destroyed its data. Based on such a warning, the owner would replicate the endangered data, therefore maintaining or even increasing storage reliability to his advantage.
- *Retrieval phase:* the owner retrieves its data from the *r* holders. If one holder destroyed the data, the owner decides on potential action towards that holder depending on its strategy (see next paragraph "Evolutionary game").

Data storage is a long-term process during which several peers may have been storing data from multiple owners; we define the evolutionary game that models our P2P storage application as a sequence of a random number of such simultaneous one-stage interactions.

Evolutionary game: Our proposed game is similar to the game in [2] where players have either the role of the donor or the role of the recipient. The donor can confer a benefit *b* to the recipient, at a cost *-c* to the donor. We consider three roles in our game: owner, holder, and verifier; any peer may play several of these roles throughout the game. In a one-stage game, the owner is considered a recipient, the *r* holders and *m* verifiers are donors. The owner gains *b* if at least one holder donates at a cost –*c*; however if no holder donates then the owner gains *βb* if at least one verifier donates at a cost –*αc* ($\alpha \leq 1$) for each verifier. The latter case corresponds to the situation where the cooperative verifier informs the owner of the data destruction, and then the owner may replicate its data elsewhere in the network thus maintaining the security of its data storage.

Holders and verifiers have the choice between cooperating, which we call interchangeably donate, or defecting:

- Cooperation whereby the peer is expected to keep others' data in its memory and to verify data held by other peers on behalf of the owner.
- Defection whereby the peer destroys the data it has accepted to hold, and also does not verify others' data as it promised to.

The peers' strategies that we consider for study are:

- Always cooperate (*AllC*): the peer always decides to donate, when in the role of the donor.
- Always defect (*AllD*): the peer never donates in the role of the donor.
- Discriminate (*D*): the discriminator donates under conditions: if the discriminator does not know its co-player, it will always donate; however, if it had previously played with its co-player, it will only donate if its co-player donates in the previous game. This strategy resembles Tit-For-Tat but differs from it in that both the owner (the donor) and its verifiers may decide to stop cooperating with the holder in the future.

3.2 Observations

Let's consider a scheme (see Fig. 1) inspired from epidemic models which categorize the population into groups depending on their state [12]. Two states are distinguished: "not known" and "known" states. Because of the random selection of holders and verifiers among all peers and given the presence of churn, there are always nodes potentially in the "not known" state.

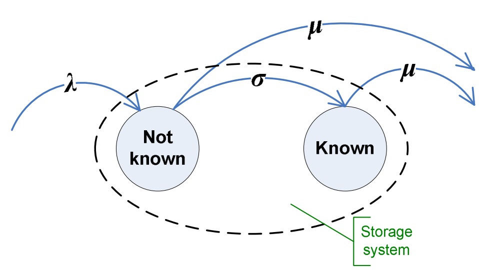

Fig. 1. System dynamics

We denote the number of peers that a given peer in average does not know at a certain time *t* by *D* and the number of peers that it knows on average at time *t* by *K*.

Peers that may join the system are peers who were invited by other members with a fixed invitation rate λ. Peers are leaving the system with a fixed departure rate of μ. The rate σ designates the frequency of encounter between two peers, one of them being the holder (i.e., the probability that a peer knows about the behavior of another peer).

We denote the total number of peers in the storage system - excluding the observing peer - as $n = D + K$. The dynamics of K and D are given by the following equations:

$$\frac{dD}{dt} = \lambda n - (\sigma + \mu)D$$

$$\frac{dK}{dt} = \sigma D - \mu K = \sigma n - (\sigma + \mu)K$$

Since $n = D + K$:

$$\frac{dn}{dt} = (\lambda - \mu)n$$

Let q be the probability that the discriminator knows what a randomly chosen co-player chose as a holder strategy in a previous one-stage game (the discriminator being an owner or verifier in that game). The probability q is equal to K/n, hence:

$$\frac{dq}{dt} = \frac{dK/dt}{n} - \frac{Kdn/dt}{n^2}$$

Thus,

$$\frac{dq}{dt} = \sigma - (\sigma + \lambda)q$$

At time t=0, the set of peers in state K is empty. Over time, peers in state D enter state K with rate σ. A new peer joining the system is assigned state D meaning that initially $q(0)$=0. The result of the above differential equation is thus:

$$q(t) = \frac{\sigma}{\sigma + \lambda}(1 - e^{-(\sigma+\lambda)t})$$

The limit of $q(t)$ when $t \to \infty$ is $\sigma/(\sigma + \lambda)$. If we consider a system without churn (λ=0), the limit becomes 1.

3.3 Fitness

We respectively denote the frequency (i.e., fraction in the population of playing peers) of strategies *AllC* by x, *AllD* by y, and D by z. The expected values for the total payoff obtained by the three strategies are denoted by U_{AllC}, U_{AllD} and U_D, and the average payoff in the population by:

$$\overline{U} = x \times U_{ALLC} + y \times U_{ALLD} + z \times U_D$$

The average payoffs that are also called fitness for each strategy are defined in the following.

At time t, a participating peer will have r times more chances to be chosen as a holder and m times more chances to be chosen as verifier than to be chosen as an owner.

A peer playing the strategy *ALLC* will always cooperate: it will donate at a cost $-c$ if it is chosen as a holder or at a cost $-\alpha c$ if it is chosen as a verifier. It will gain a benefit b if it is chosen as an owner and at least one of its data holders is not a defector, otherwise, it may gain a benefit βb if at least one of its verifiers is not a defector.

$$U_{ALLC} = -rc - m\alpha c + b(1 - y^r) + \beta b\big(y^r(1 - y^m)\big)$$
$$= -c(r + m\alpha) + b(1 - y^r + \beta y^r(1 - y^m))$$

A peer playing the strategy *ALLD* will never cooperate, so it will never donate. It will gain a benefit b if it is chosen as an owner and at least one of its data holders is not any of these types: a defector (type occurs with frequency, i.e., probability y) or a discriminator that knows the peer (type occurs with probability qz on average). Otherwise, the peer may gain a benefit βb if at least one of its verifiers is not of any of the former two types.

$$U_{ALLD} = b(1 - (y + qz)^r) + \beta b\big((y + qz)^r(1 - (y + qz)^m)\big)$$
$$= b(1 - (y + qz)^r + \beta(y + qz)^r(1 - (y + qz)^m))$$

A peer playing the strategy D will always cooperate if it does not know the recipient or the latter was cooperative in a previous interaction. It will donate at a cost $-c$ if it is chosen as a holder or at a cost $-\alpha c$ if it is chosen as a verifier. It will gain a benefit b if it is chosen as an owner and at least one of its data holders is not a defector, otherwise, it may gain a benefit βb if at least one of its verifiers is not a defector.

$$U_D = -c(r + m\alpha)(1 - qy) + b(1 - y^r + \beta y^r(1 - y^m))$$

Strategies with higher fitness are expected to propagate faster in the population and become more common. This process is called *natural selection*.

3.4 Replicator dynamics

The basic concept of replicator dynamics is that the growth rate of peers taking a strategy is proportional to the fitness acquired by the strategy. Thus, the strategy that yields more fitness than the average fitness of the whole system increases, and vice versa. We will use the well known differential replicator equations:

$$\frac{dx}{dt} = x(U_{ALLC} - \overline{U})$$
$$\frac{dy}{dt} = y(U_{ALLD} - \overline{U})$$
$$\frac{dz}{dt} = z(U_D - \overline{U})$$

3.5 Evolutionary stable strategy

A Strategy is said to *invade* a population of strategy players if its fitness when interacting with the other strategy is higher than the fitness of the other strategy when interacting with the same strategy. An evolutionarily stable strategy (ESS) is a strategy which no other strategy can invade if all peers adopt it.

Case $x \neq 0$, $y = 0$, $z \neq 0$: This case corresponds to a fixed point in the replicator dynamics, which means that a mixture of discriminating and altruistic population can coexist and are in equilibrium.

Case *x*≠0, *y*≠0, *z*=0: In this case, the replicator dynamics of both altruistic and defector populations are:

$$\frac{dx}{dt} = -xyc(r + m\alpha) \leq 0$$
$$\frac{dy}{dt} = xyc(r + m\alpha) \geq 0$$

The population of defectors wins the game and the ESS is attained at *x*=0 and *y*=1.

Case *x*=0, *y*≠0, *z*≠0: There is an equilibrium point for which defectors and discriminators coexist (*x*=0, $y=y_0$, $z=z_0$) which corresponds to:

$$\frac{dy}{dt} = \frac{dz}{dt} = 0$$

The equilibrium point is then solution of the following equation:

$$c(r + m\alpha)(1 - qy_0) = b\Big((y_0 + qz_0)^r - {y_0}^r + \beta\big({y_0}^r(1 - {y_0}^m) - (y_0 + qz_0)^r(1 - (y_0 + qz_0)^m)\big)\Big)$$

Table 1 describes equilibrium values in some particular cases. More cases for equilibrium values will be examined in Section 4.

Table 1. Finding the equilibrium for *x*=0, *y*≠0, *z*≠0.

Conditions	y_0	z_0
r=1, *m*=0, *b*≠*c*, $q(t) \xrightarrow[t\to\infty]{} \frac{\sigma}{\sigma+\lambda}$	$\min\left(\max\left(\frac{b\sigma - c(\sigma+\lambda)}{(b-c)\sigma}, 0\right), 1\right)$	$\min\left(\max\left(\frac{c\lambda}{(b-c)\sigma}, 0\right), 1\right)$
r=0, *m*=1, *b*≠*c*, $q(t) \xrightarrow[t\to\infty]{} \frac{\sigma}{\sigma+\lambda}$	$\min\left(\max\left(\frac{\beta b\sigma - \alpha c(\sigma+\lambda)}{(\beta b - \alpha c)\sigma}, 0\right), 1\right)$	$\min\left(\max\left(\frac{\alpha c\lambda}{(\beta b - \alpha c)\sigma}, 0\right), 1\right)$
r=1, *m*=1, $q(t) \xrightarrow[t\to\infty]{} 1$	$\min\left(\max\left(\frac{c(1+\alpha) - b}{\beta b}, 0\right), 1\right)$	$\min\left(\max\left(\frac{(1+\beta)b - c(1+\alpha)}{\beta b}, 0\right), 1\right)$

Case *x*≠0, *y*≠0, *z*≠0: There is one stationary point (*x*=0, $y=y_0$, $z=z_0$) for which defectors will exploit and eventually deplete all cooperators. The amount of defectors will first increase, and then converges to the equilibrium where there is either coexistence with discriminators, or winning over them, or losing to them depending on storage system parameters.

4 Numerical evaluation

The evolutionary game is simulated with a custom simulator using the differential equations of subsection 3.4, and within several scenarios varying storage system's parameters.

Initial frequency of strategies: Fig. 2 shows the frequency of cooperators and defectors over time, and demonstrates that with time cooperators will be eliminated from the system by these defectors. The presence of discriminators in the system does not prevent cooperators from being evicted from the system; however, discriminators and defectors will converge to an equilibrium where both coexist (see Fig. 3).

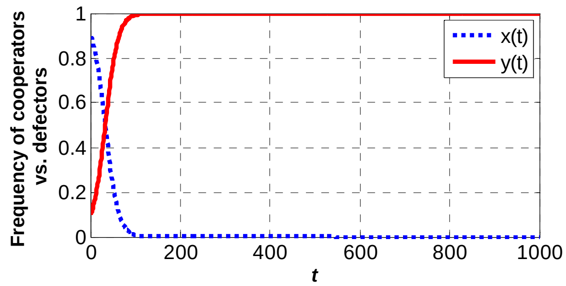

Fig. 2. Frequency of cooperators vs. defectors over time. m=5, r=7, β=0.1, α=0.001, λ=0.01, σ=0.05, b=0.05, c=0.01, $x(0)$=0.9, $y(0)$=0.1, and $z(0)$=0.

This equilibrium is perturbed by the injection of a large population of defectors, as illustrated in Fig. 4 (by varying the initial frequency of z). If discrimination becomes a minor strategy in the population (frequency $\leq$ 0.1), it is completely eliminated from the system. However, if a small population of defectors is injected, discriminators converge to the same equilibrium. So, there are two equilibria that are determined by the initial population of discriminators: (x=0, y=1, z=0) and (x=0, y=y_0, z=z_0).

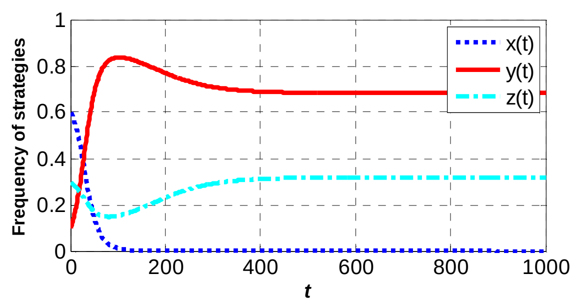

Fig. 3. Frequency of the three strategies over time. m=5, r=7, β=0.1, α=0.001, λ=0.01, σ=0.05, b=0.05, c=0.01, $x(0)$=0.6, $y(0)$=0.1, and $z(0)$=0.3.

The discriminators do not win over defectors, because the latter may still have a good payoff if they interact with some discriminators that do not know them yet, for instance for discriminators that just entered the system, or defectors that just joined in. Additionally, defectors do not always win over the discriminators because there are discriminators that already know them and that always choose to defect with them. The figure shows also a little decrease in the frequency of discriminators before converging to the equilibrium. The decrease is due to the fact that discriminators act as cooperators in the beginning of the game since they do not know the behavior of defectors yet.

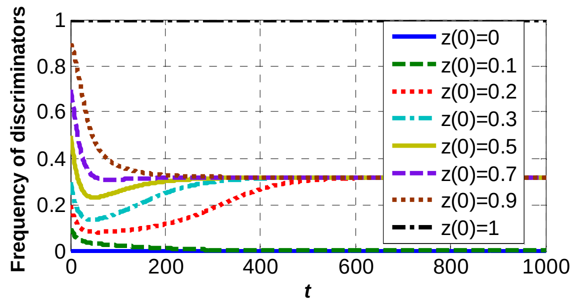

Fig. 4. Frequency of discriminators over time varying $z(0)$. m=5, r=7, β=0.1, α=0.001, λ=0.01, σ=0.05, b=0.05, c=0.01, and $x(0)$=0.

Number of verifiers and replicas: Varying the number r of data replicas or the number m of verifiers changes the equilibrium point. Increasing r favors defectors (see Fig. 5a).

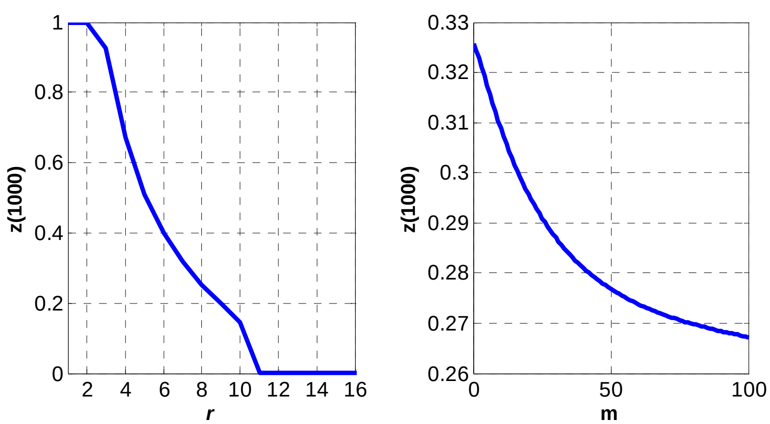

Fig. 5. Frequency of discriminators at equilibrium varying (a) r and m. r=7, m=5, β=0.1, α=0.001, λ=0.01, σ=0.05, b=0.05, c=0.01, $x(0)$=0, $y(0)$=0.5, and $z(0)$=0.5.

This is because the fitness gain of discriminating owners is overwhelmed by the fitness loss that results from data storage cost -*c* that is always paid by discriminating holders. Increasing *r* increases data reliability, thus increasing chances of having the benefit *b*. But, this benefit is perceived by both populations of discriminators and defectors without favoring one over the other.

Increasing *m* decreases the equilibrium value of discriminators frequency (see Fig. 5b). This is due to the fact that increasing *m* increases the cost of data verification $-\alpha c$. This cost is just paid by discriminating verifiers. That's why increasing *m* reduces their fitness. Fig. 6 also illustrates the fact that increasing the probability of encounter σ leads to an increase in the equilibrium value of discriminators' frequency because more discriminators get acquainted with more defectors.

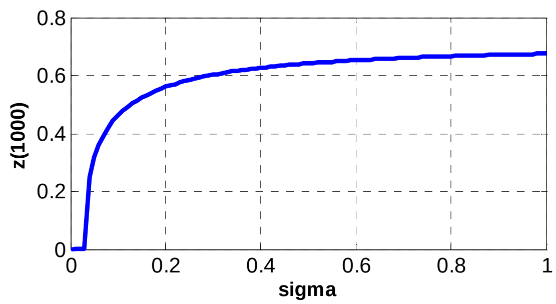

Fig. 6. Frequency of discriminators at equilibrium varying the probability of encounter σ. m=5, r=7, β=0.1, α=0.001, λ=0.01, b=0.05, c=0.01, $x(0)$=0, $y(0)$=0.5, and $z(0)$=0.5.

Churn: The peer arrival rate λ affects the probability *q*, and hence the equilibrium point of the game (see Fig. 7). For a low churnout value (small λ), the frequency of discriminators at equilibrium is high; whereas for a high churnout value (large λ) the frequency at equilibrium decreases. For high churnout, peers are not able to get acquainted with all peers since there are always new peers in the system, and defectors may take advantage of the lack of knowledge of discriminators about the system to gain benefit and remain in the game. For a system without churnout (λ=0), discriminators win against defectors that are eliminated from the game.

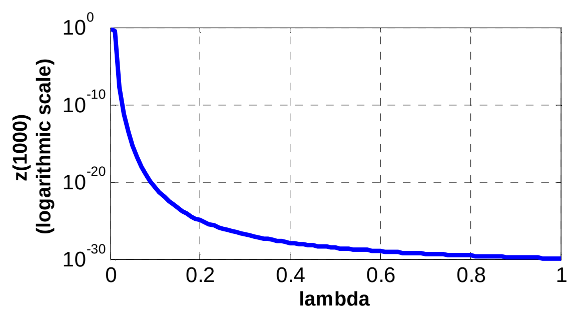

Fig. 7. Frequency of discriminators at equilibrium varying the arrival rate λ. m=5, r=7, β=0.1, α=0.001, σ=0.05, b=0.05, c=0.01, $x(0)$=0, $y(0)$=0.5, and $z(0)$=0.5.

Benefit and cost: Fig. 8 depicts the impact of the benefit *b* and of the cost *c* on the frequency of discriminators at equilibrium. The figure shows that *b* and *c* have opposite effects on the equilibrium frequency of discriminators: increasing *b* increases the frequency whereas increasing *c* makes it decrease. If the storage cost is small, it will be compensated by the benefit. In contrast, if the storage cost is high ($c \geq 0.3 \times b$), discriminators cannot cope with this high cost and they will be eliminated from the system by defectors.

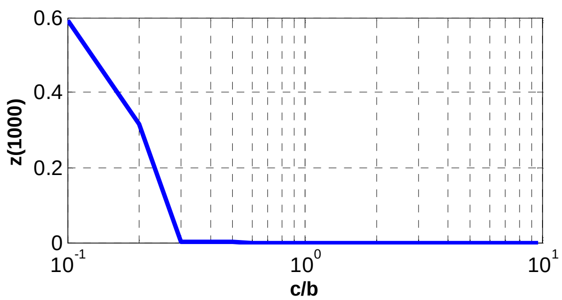

Fig. 8. Frequency of discriminators at equilibrium varying the ratio *c*/*b*. m=5, r=7, β=0.1, α=0.001, λ=0.01, σ=0.05, b=0.05, $x(0)$=0, $y(0)$=0.5, and $z(0)$=0.5.

Summary: Simulation results prove that there exist parameter values for which discriminators, who use an audit-based mechanism, may win against free-riding defectors. Discriminators are not hopeless when confronting defectors, even if the latter may dominate altruists (*always cooperate* strategy). At the equilibrium of the game, both discriminators and defectors may coexist if there is churn in the system otherwise discriminators will dominate. The replication rate *r* and the number of verifiers *m* decreases the frequency of discriminators at the equilibrium for a small value for *r*. Additionally, a costly storage or an increase in the number of verifications performed reduce this frequency.

5 Conclusion

In this paper, we validated an audit-based strategy as an evolutionary stable strategy under some conditions as the basis for a P2P storage system. The results obtained show that such a strategy wins over a free-riding strategy in a closed system. Given reasonable constraints, they also show that this strategy can coexist with free-riders, and even achieve a high frequency. The fact that cryptographic primitives exist that make the implementation of appropriate audit mechanisms possible without unrealistic network bandwidth requirements is noteworthy for practical implementation. We are currently investigating other security issues of P2P storage systems, notably those raised by maliciousness.

Acknowledgements

The work presented here has been partially funded by the project SPREADS (ANR) and the ReSIST IST-026764 NoE.

References

1. Daniel Friedman, “On economic applications of evolutionary game theory”. *Journal of Evolutionary Economics* 8, pp. 15–43.
2. Hannelore Brandt and Karl Sigmund, “The good, the bad and the discriminator--errors in direct and indirect reciprocity”, *Journal of Theoretical Biology*, Volume 239, Issue 2, 21 March 2006, Pages 183-194.
3. Nouha Oualha, Melek Önen, and Yves Roudier, “A Security Protocol for Self-Organizing Data Storage”, in Proceedings of 23rd International Information Security Conference (SEC 2008) *IFIP Sec 2008*, September 2008, Milan, Italy.
4. Nouha Oualha, Melek Önen, and Yves Roudier, “A Security Protocol for Self-Organizing Data Storage”, (extended version) Technical Report Nº RR-08-208, EURECOM, January 2008.
5. N. Oualha, Y. Roudier, “Securing ad hoc storage through probabilistic cooperation assessment”, WCAN '07, 3rd Workshop on Cryptography for Ad hoc Networks, July 8th, 2007, Wroclaw, Poland.
6. G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z. Peterson and D. Song, “Provable data possession at untrusted stores”CCS '07: *Proceedings of the 14th ACM conference on Computer and communications security, ACM, 2007*, 598-609.
7. A. Juels and B. S. Kaliski, “PORs: Proofs of retrievability for large files”, Cryptology ePrint archive, June 2007. Report 2007/243.
8. Y. Deswarte, J.-J. Quisquater, and A. Saïdane, “Remote Integrity Checking”, In *Proceedings of Sixth Working Conference on Integrity and Internal Control in Information Systems (IICIS)*, 2004.
9. D. G. Filho and P. S. L. M. Barreto, “Demonstrating data possession and uncheatable data transfer”, In *IACR Cryptology ePrint Archive*, 2006.
10. Francesc Sebe, Josep Domingo-Ferrer, Antoni Martínez-Ballesté, Yves Deswarte, Jean-Jacques Quisquater, “Efficient Remote Data Possession Checking in Critical Information Infrastructures,” *IEEE Transactions on Knowledge and Data Engineering*, 06 Aug 2007. IEEE Computer Society Digital Library. IEEE Computer Society, 6 December 2007 http://doi.ieeecomputersociety.org/10.1109/TKDE.2007.190647
11. T. Schwarz, and E. L. Miller, “Store, forget, and check: Using algebraic signatures to check remotely administered storage”, In *Proceedings of the IEEE Int'l Conference on Distributed Computing Systems (ICDCS '06)*, July 2006.
12. Douglas Samuel Jones and B. D. Sleeman, “Differential Equations and Mathematical Biology”, London: Allen & Unwin, 1983.
13. Nouha Oualha and Yves Roudier, “Reputation and Audits for Self-Organizing Storage”, *Workshop on Security in Opportunistic and SOCial Networks SOSOC 2008*, September 2008, Turkey, Istanbul.
14. Nouha Oualha, Pietro Michiardi, and Yves Roudier, “A game theoretic model of a protocol for data possession verification”, *IEEE International Workshop on Trust, Security, and Privacy for Ubiquitous Computing TSPUC 2007*, June 18, 2007, Helsinki, Finland.