Institut Eurécom
Corporate Communications Department
2229, route des Crêtes
B.P. 193
06904 Sophia Antipolis
FRANCE

Research Report RR-08-222

A Reputation System for Self-Organizing Storage

19 May 2008

Nouha Oualha and Yves Roudier[1]

Tel: (+33) 4 93 00 81 00
Fax: (+33) 4 93 00 82 00
Email: {oualha, roudier}@eurecom.fr

[1] Eurecom's research is partially supported by its industrial partners: BMW Group Research & Technology – BMW Group Company, Bouygues Telecom, Cisco Systems, France Telecom, Hitachi, SFR , Sharp, STMicroelectronics, Swisscom, Thales

A Reputation System for Self-Organizing Storage

Nouha Oualha and Yves Roudier

Abstract

Reputation systems have demonstrated their interest in stimulating cooperation in peer-to-peer (P2P) systems, even though they are susceptible to collusion and bashing. In addition, computing reputation generally relies on a partial assessment of the behavior of peers only, which might delay the detection of selfish peers. This situation is rendered even worse in self-organized storage applications, since storage is not an instantaneous operation and data are vulnerable throughout their entire storage lifetime. This paper introduces a local reputation system for P2P storage in which peer observations are carried out through the periodic verification of a proof of data possession, and which in addition addresses the aforementioned issues of reputation systems.

Index Terms

Trust establishment, reputation, peer-to-peer, cooperation

Contents

List of Figures

1. Introduction

Peer-to-Peer (P2P) systems have emerged as an important paradigm for distributed storage in the way they exploit and efficiently make use of untapped peers' storage resources. Particularly motivating services for P2P data storage are AllMyData [1], Wuala [2], and Ubistorage [3] where data is outsourced from the data owner place to several heterogonous storage sites in the network, for increased data availability and fault-tolerance, reduced storage maintenance costs, and high scalability.

P2P data storage essentially means that a data *owner* peer stores its data at a third-party *holder* peer which is supposed to faithfully store the very data and make them available to the owner (and perhaps others) on demand. Since such P2P storage systems thrive on free storage space, a major security-related issue associated with them is how to incite peers to concede some of their spare storage space in favor of other peers, and at the mean time how to efficiently and fairly ensure that a peer who grants usage of some of its own space to store other peers' data is normally granted usage of a proportional amount of space somewhere else in the network, for his own data storage.

Approaches inciting peer cooperation and ensuring secure storage and storage fairness are generally based on reputation. The reputation value of a peer is an evaluation of its past behavior used by other peers to evaluate how trustful it is. In a data storage application, peers favor storing their personal data at well famed siblings. Peers with a bad reputation are on the contrary gradually isolated from the storage system.

Most approaches to building reputation systems are making simplifying assumptions on the instantaneous propagation of indirect reputation information around the system and on the willingness of peers to correctly and fairly propagate such information. We propose in this paper a new reputation mechanism that relies only on direct observations thereby serving a twofold objective: inciting peers to check the availability of others' data and at the same time estimating reputation based on the very results of verification.

The remainder of the paper is organized as follows: Section 2 gives an overview of the P2P data storage we are intending to enhance with a reputation mechanism, and presents the attacks that this system is exposed to. Section 3 describes the reputation mechanism and particularly proves the satisfactory use of direct observations in estimating reputation values. Section 4 discusses implementation issues of the reputation mechanism on top of a P2P storage system, notably regarding the mitigation of denial of service attacks on the reputation mechanism. Section 5 validates the ability of the reputation mechanism to filter out selfish peers from the storage system and to improve the availability of stored data. Section 6 covers related work. Section 7 finally presents our concluding remarks.

2. P2P storage: An overview

A P2P storage application allows *owner* peers to store their personal data in replicas at several *holder* peers. A stored data replica is periodically checked by *verifier* peers on behalf of the owner. The verification process relies on a secure data possession verification protocol as discussed in [4] and [5]. Peers interact with each other based on trust relationships that are established through reputation: the higher the reputation of a peer, the more trustworthy and reliable it is believed to be.

2.1. Data storage

The storage of data in the system relies on several phases:

- **Verification delegation:** The owner delegates the task of verifying data stored in the system to well reputed peers.
- **Data storage:** The owner stores *r* data replicas at peers that are selected with the help of verifiers.
- **Verifier checking:** Each verifier checks the storage at the holder using a secure data possession verification protocol. With the result of this checking, the verifier updates its estimate of the reputation value of the holder.
- **Owner checking:** The owner receives verification results from all verifiers. It checks the consistency of these results: if more than half of the verifiers agree on the same result, it accepts that result as the correct one; however, if there is no dominant result, the owner will ultimately and opportunistically check the availability of its data at the holder by itself. With this a posteriori checking, the owner decides if it must again replicate its

data in the system with new holders, and at the same time it updates the reputation values of the checked holders and of the verifiers.

- **Data retrieval:** The owner retrieves its data from holders, which frees them from their obligations. This operation may be assisted by verifiers to ensure that data are actually sent back to the owner.

2.2. Adversary model

The adversaries that we consider for such application are peers that do not correctly follow the roles (owner, data holder, or data verifier) that they agreed to carry out, and trick the reputation system for any perceived personal benefit: they seek to use the system storage without contributing their fair share, or intentionally attack other peers or their storage in the system. In the following, we examine ways which peers may use to subvert the reputation-based P2P storage system.

Storage related attacks:

- **Free-riding:** free-riders are peers that do not contribute to the stores community, or that may destroy some data they promised to keep in order to optimize their own storage resources.
- **Collusion between holders:** Holders collude so that only one of them keep data replica, and the remainder of holders are still able to answer challenges to verifiers by invoking the holder with the replica, and hence increase their reputation at these verifiers. This collusion is mitigated by personalizing data replicas stored at different holders as proposed in [4] and [5].
- **Maliciousness:** Malicious peers aim at destroying either data or the infrastructure with DoS attacks (e.g., flooding), even at the expense of their own resources. Maliciousness can be prevented using common security countermeasures for DoS attacks.

Reputation related attacks:

- **Lying:** a liar is a peer that disseminates incorrect observations on other peers (*rumor spreading*) in order to either increase or decrease their reputation. Colluded liars may form a collective of peers that conspires against one or more peers in the network by assigning unfairly low reputation to them (*bad mouthing*) and high reputation for themselves.
- **Collusion between owner and holder:** The collusion aims at increasing the reputation of the holder at honest verifiers. Just lying to verifiers supposes that observations of peers rely on external recommendations. However without these recommendations, peers may still be vulnerable to lying using such type of collusion where the owner pretends storing bogus data at the holder.
- **Collusion between holder and verifier:** The aim of such collusion is to advertise the quality of holder more than its real value (*ballot stuffing*) thus increasing its reputation at owner. But, still the owner may ultimately and opportunistically check by itself storage at holder to make its own view on the holder.
- **Sybil attack:** If peers are able to generate new identities at will, they may use some of them to increase the reputation of the rest of identities either by lying, or pretending to have several roles at the same time.

3. Reputation mechanism

We propose a new reputation system for P2P storage applications that allows estimating the trustworthiness of peers based on experiences and observations of their past behaviors. In the following, the different features of our reputation system are thoroughly described.

3.1. Behavior observations

The reputation of a peer is estimated based on the observation of its behavior by third parties. The semantics of the information collected can be described in terms of direct (or local) or indirect (or system-wide) observations. Direct observation amounts to the compilation of a history of personal interactions by one peer towards another peer when being the owner of data stored at the peer or serving as verifier of this peer. On the other hand, indirect observation refers to any reputation information received from other peers in the system. There are substantial communication savings to be gained by limiting observations to just private interactions even though indirect observation may be only partially disseminated or piggybacked on ordinary messages. Besides, using only direct observation may delay the evolution of reputation.

3.1.1 Analytic model

This section discusses how to compute the gain of choosing one way of observation reciprocity over the other in terms of the level of correctness of gathered reputation information.

Considering two peers p_1 and p_2, where p_1 desires to have correct observations on p_2. Peer p_1 may perform a correct observation itself or may receive observations from other peers in the system that may be correct or incorrect. Our model assumes that incorrect observations are received from dishonest peers only. Let's η denote the fraction of dishonest peers in the total population.

We define a quality level for the estimated observation with two extrema: $\overline{o}$ and $\underline{o}$. An observation of quality $\overline{o}$ is correct, and an observation of quality $\underline{o}$ is incorrect. Observation may be null to refer to the situation where p_1 does not have any observation on peer p_2 (indistinguishably from the worst reputation).

First of all, the probability that p_1 knows about the p_2's behavior is computed (it must at least obtain the result of one interaction involving p_2); the estimated observation of p_1, denoted $\tilde{o}$, is then derived for three different cases:

- Observations based only on storage results: p_1 only takes into account its personal interactions with p_2 as an owner peer storing data at p_2.
- Adding observations based on verifications results: p_1 only takes into account its personal interactions with p_2 as an owner storing data at p_2 or as a verifier for other peers' data stored at p_2.
- Adding observations based on peer recommendations: p_1 takes into account both its personal interactions and opinions expressed by other peers with respect to p_2.

Observations based only on storage: The probability that p_1 knows about the behavior of p_2 is equal to:

$$\Pr ob[p_1 \, knows \, p_2] = \theta_1 = \lambda \times r/(n-1)$$

λ being the average storage rate of peers and n being the number of peers.

Since personal observations are always correct, the estimated observation quality may only have two values: a correct observation or no observation.

$$\Pr ob[\tilde{o}_1 = \overline{o}] = \theta_1$$
$$\Pr ob[\tilde{o}_1 = \underline{o}] = 0$$
$$\Pr ob[\tilde{o}_1 = 0] = 1 - \theta_1$$

On average, we have:

$$\tilde{o}_1 = \theta_1 \times \overline{o}$$

Direct observations based on storage or verification: The probability that p_1 knows about the behavior of p_2 is equal to:

$$\Pr ob[p_1 \, knows \, p_2] = \theta_2 = 1 - (1-\theta_1) \times$$
$$(1 - \theta_1 + \theta_1 \times (1 - m/(n-2))^r)^{n-2}$$

The estimated observation quality may have two values: correct observation or no observation.

$$\Pr ob[\tilde{o}_2 = \overline{o}] = \theta_2$$
$$\Pr ob[\tilde{o}_2 = \underline{o}] = 0$$
$$\Pr ob[\tilde{o}_2 = 0] = 1 - \theta_2$$

On average, we have:

$$\tilde{o}_2 = \theta_2 \times \overline{o}$$

Including observations based on peer recommendations: The probability that p_1 knows about the behavior of p_2 is equal to:

$$\Pr ob[p_1 \, knows \, p_2] = \theta_3 = 1 - (1-\theta_2)^{\gamma \times n}$$

γ being the fraction of the peer population to which the reputation is propagated.

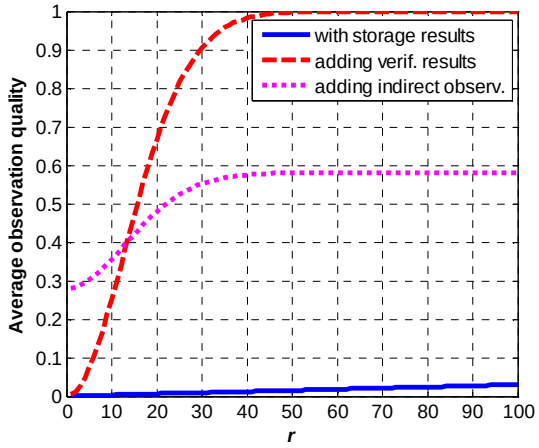

Figure 1 Average observation quality: (a) varying *r* and (b) varying *m*. *n*=1000, *λ*=0.1, *γ*=0.3, *m*=10, *w*=0.3, *η*=0.3.

External observations may either originate from honest peers or from dishonest peers. Peer p_1 receives at best $(1-\eta)\times\gamma\times n$ observations from honest peers and $\eta\times\gamma\times n$ from dishonest peers. Observations from honest peers are all correct; and observations from malicious peers are always incorrect. For k and k' not null observations respectively received from honest and dishonest peers, the average observation quality is denoted by $t_{k,k'}$ when p_1 has a direct observation, and by $t'_{k,k'}$ when p_1 does not have a direct observation:

$$t_{k,k'} = (1-w)\bar{o} + w(k\bar{o} + k'\underline{o})/(k+k')$$

$$t'_{k,k'} = w(k\bar{o} + k'\underline{o})/(k+k')$$

w being the weight that p_1 gives to averaged system-wide observations with respect to local observations. For $0 \leq k \leq (1-\eta)\times\gamma\times n$ and $0 \leq k' \leq \eta\times\gamma\times n$, we have:

$$\Pr ob[\tilde{o}_3 = t_k] = (C^k_{(1-\eta)\gamma n}\theta_2{}^{k+1}(1-\theta_2)^{(1-\eta)\gamma n-k}) \times (C^{k'}_{\eta\gamma n}\theta_2{}^{k'}(1-\theta_2)^{\eta\gamma n-k'})$$

$$\Pr ob[\tilde{o}_3 = t'_k] = (C^k_{(1-\eta)\gamma n}\theta_2{}^{k}(1-\theta_2)^{(1-\eta)\gamma n-k+1}) \times (C^{k'}_{\eta\gamma n}\theta_2{}^{k'}(1-\theta_2)^{\eta\gamma n-k'})$$

$$\Pr ob[otherwise] = 0$$

The value $C^k_{(1-\eta)\times\gamma\times n}$ (respectively $C^{k'}_{\eta\times\gamma\times n}$) is the number of combinations of k (respectively k') peers from the set of honest (respectively dishonest) peers from which p_1 gathers observations. A certain probability of interaction is attached to the observations of both honest and dishonest peers. This is due to the fact that even though peers have to provide cryptographic proofs that they had interactions with p_2, even honest peers cannot always provide proofs of correct observation: for example, the observation of the absence of any response from p_2 cannot be proved; or the peer sending an observation may be in collusion with p_2.

Using the Vandermonde's identity, we have on average:

$$\tilde{o}_3 = \theta_2(1-w) + w((1-\eta)\times\overline{o} + \eta\times\underline{o})$$

Comparison: Seeking for simplicity, we choose quality observations such as: $\overline{o} = 1, \underline{o} = -1$. Thus, we have:

$$\tilde{o}_1 = \theta_1$$

$$\tilde{o}_2 = \theta_2$$

$$\tilde{o}_3 = \theta_2(1-w) + w(1-2\eta)$$

The average quality of observations is computed in the three cases. Figure 1 demonstrates that the best quality is obtained in the second case where all direct observations are taken into account which is the choice we made in the reputation mechanism proposed in this paper. This is obtained for a minimum replication rate (r>15 for m=10) and a minimum number of verifiers per replica (m>40 for r=7).

If the ratio of maliciousness in the system increases, the quality of observation in the case with indirect observations linearly decreases with this ratio, however this quality is not affected in case of direct observations, as it is depicted in Figure 2.

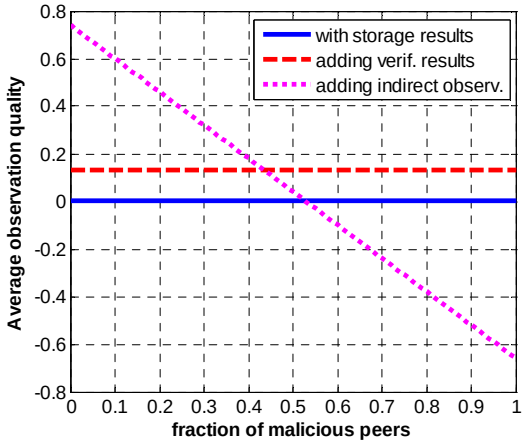

Figure 2 Average observation quality varying the fraction of malicious peers. n=1000, λ=0.1, γ=0.3, r=7, m=10, w=0.3.

Figure 3 shows that increasing the peers' population number n leads to a decrease in the quality of observations in the three cases, especially cases with the verifications results taken into account. For a large population, indirect observations case becomes more advantageous in terms of observation quality than the case with just direct observations.

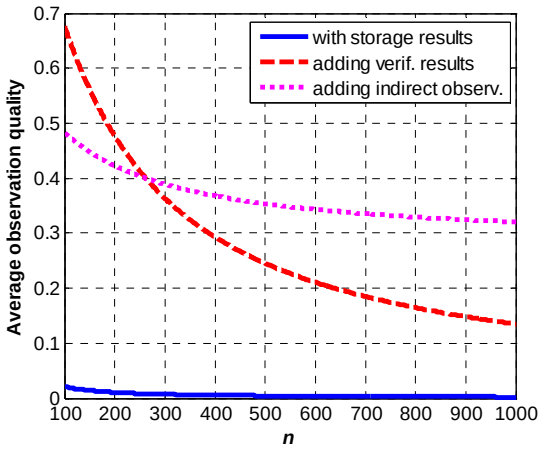

Figure 3 Average observation quality varying the number of peers. λ=0.1, γ=0.3, r=7, m=10, w=0.3, η=0.3.

3.1.2 Observations in our reputation mechanism

In our reputation mechanism, the estimation of reputation solely relies on direct observations: either between a verifier and its assigned holder or between a data owner and its data holder. The verifier checking storage at some holder can estimate the degree of cooperation of this holder. Using verifications results sent back by verifiers, the owner can estimate the likelihood of its data availability at the very holder. These estimations are used to compute personal reputation values towards a given holder, and are locally stored at each peer in a reputation list. No propagation of reputation is necessary which avoids the problem of liars and also the effect of rumor spreading.

3.2. Reputation computation

Our reputation function is based on the simple model of Linear Increase Sudden Death (LISD) (see Figure 4): when a peer observes that another peer is still keeping data stored, it linearly increases its estimate of reputation of this very peer; however, if it detects that the peer has destroyed data it has promised to store, it clears the reputation value to 0. Initially, all peers start with reputation 0. This means that selfish peers changing their identities do not gain any advantage of that, because they still have a reputation of 0. We didn't use a blacklisting model where whenever a peer destroys data, it is blacklisted by verifiers and the data owner, because the cause of data destruction or corruption may be due to a crash or fault at the peer hence blacklisting may produce a lot of false positives which may severely reduce the number of peers cooperating with the owner. Other trust models can be adopted like for example the Additive Increase Multiplicative Decrease (AIMD) mechanism.

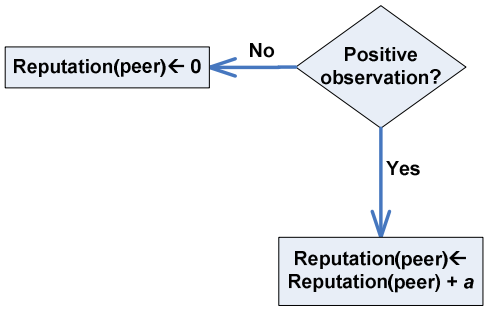

Figure 4 LISD reputation model: *a* parameter for linear increase.

A reputation value must be expressive of the expected behavior of a peer, and should be representative of its recent behavior rather than its old rating assigned to it. Therefore, a decaying factor is incorporated into the reputation function (see Figure 5).

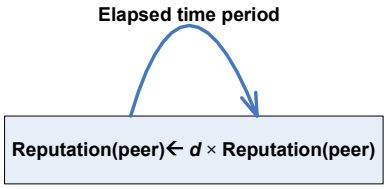

Figure 5 Decaying reputation function: *d* is the decaying factor

3.3. Interaction decision

An interaction between two peers is effective, if the reputation of each peer at each other is higher than a certain threshold value: potential holders or verifiers will accept the owner request if the reputation value of the owner is higher than this threshold. The threshold is variable over time depending on the current state of peers. An example of such threshold is the mean function of reputation values: a peer interacts with another peer if the reputation of this latter estimated by the first one is higher than the mean of all reputation values it is holding.

4. Implementing Reputation with Storage

In the P2P storage system, we rely on the construction of groups in which we evaluate peers reputation. Peers store their personal data in their group. The security of data stored is the responsibility of group members, given that they are periodically verified by some group members for availability and no corruption.

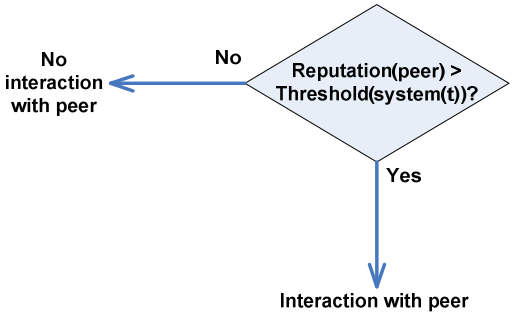

Figure 6 Interaction decision

4.1. Group construction and management

Peer groups are dynamic with members that join and leave the group at anytime. Such group-based architecture allows only intra-group interactions, and thus peers establish rapid knowledge of the trustworthiness of their group fellows. Moreover, the group ensures a minimum level of good behavior: whenever a peer misbehaves its reputation at the group decreases until a threshold at which it is ejected from the group.

Peer groups are created either in a centralized or in a decentralized manner. Centralized managed groups can be constructed at outset by an authority like partnership in [10] that may tackle as well the task of distributing the group key to all members. On the other hand decentralized groups are cooperatively formed at will by its members and

they rely on collaborative group key agreement protocols (e.g., [6], [7]). The group key controls the access to the group, and ensures secure and private communication between its members.

In the group, peers have unique identities. The risk of Sybil attacks can be mitigated by imposing a membership fee for peers willing to join a given group, or in a decentralized way constraining the number of invitations any group member possesses as proposed in [8].

4.2. Self-organizing peer selection

The reputation-based P2P storage system allows peers to store their data at well reputed peers, the holders, and also proposes to owners to delegate the verification of their data to other volunteer peers, the verifiers.

4.2.1 Verifier selection

A data owner desiring to store a data replica in the system first sends a verification request to its group members. From peers answering to this request, the owner selects *m* verifiers based on their estimated reputation level, and then acknowledges them including in the message the list of the *m* chosen verifiers. This information is a commitment from the owner to the verifiers' list. The distribution of the verification task to several verifiers mitigates the collusion between the holder and one or several verifiers (with number at least less than *m*/3).

4.2.2 Holder selection

To avoid collusion between the owner and the holder, selected verifiers will choose altogether the holder for the owner. Each verifier proposes a list of well reputed holders to the owner (see Figure 7). From these verifiers' lists and taking in account the rating order at these lists, the owner constructs a short list of potential holders. The owner sends a storage request to the holder on top of the short list. If the holder does not accept the storage request, the owner sends the same storage request to the next holder in the short list, and so on until one holder accepts to store owner's data. Then, the owner informs the verifiers of the holder that they have to check, including in the message the metadata for data verification, verifiers' lists, and the final short list. Verifiers may opportunistically check the holders that have refused to store owner's data, to be convinced that the owner did actually request them storage and they refused. With all this information, verifiers can be sure that the owner respects their choices, and that there is no premeditated collusion between the owner and the holder. If the owner does not agree with the holders' short list it may form a new one with the help of other verifiers.

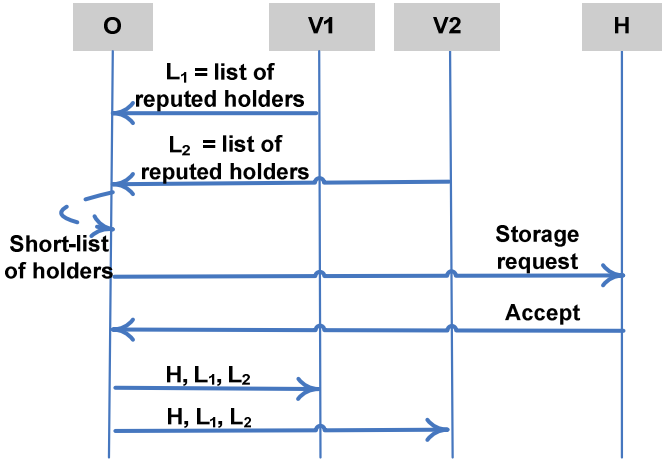

Figure 7 Holder selection: the owner O selects a holder H with the help of verifiers V1 and V2.

It is clear that the operation of holder selection requires several communication messages between the owner and verifiers that might be grouped in a single multicast message; nevertheless, this is the price to pay to obtain a

contracted agreement between owner, verifiers, and holder, and particularly to avoid collusion between any participants in this agreement.

5. Simulation experiments

To validate our reputation-based P2P storage system, we implemented a custom simulator whose framework is at first described, and then results of simulation are presented and analyzed.

5.1. Framework

The self-organizing network is a modeled as a closed set of peers with a fixed storage rate and persistent behavior (peers keep their behavior types during the simulation). We consider three behavior types:

- **Cooperation** whereby the peer concedes storage space for other peers' data and sends correct verification results to owner.
- **Passive selfishness** whereby the peer free rides by using the storage offered in the network without contributing its equal share.
- **Active selfishness** whereby the peer only probabilistically conserves data stored and gives incorrect verification results to the owner with some probability.

5.2. Simulation results

The framework is simulated in different scenarios in order to analyze the impact of system parameters and choices on the convergence time of the storage system to a stable state where cooperative peers store their data at cooperative peers.

Filtering out of selfish peers. Figure 8 shows the composition of holders in terms of their behavior types with time. In order to pick the right holder for the owner, verifiers' choices are sufficient to distinguish cooperative peers from selfish ones, and allow the selection of a cooperative holder rather than a selfish one. After less than 20 simulation cycles (cycle = time verification period), cooperative holders are 90% of the total holders in the storage system.

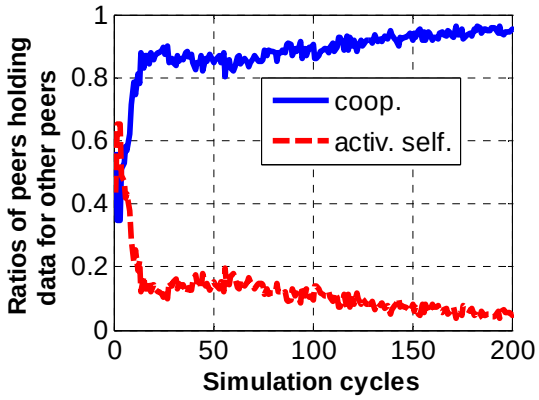

Figure 8 Ratios of cooperative and selfish peers holding data for other peers. λ=0.5, n=100, r=7, m=5, η=0.5 (0.4 active selfishness, 0.1 passive selfishness).

Figure 9 shows that after just 10 simulation cycles, 90% of owners are cooperative peers. This demonstrates that cooperative peers are very active for storing data, more than selfish ones that are gradually denied the storage of their data in the system.

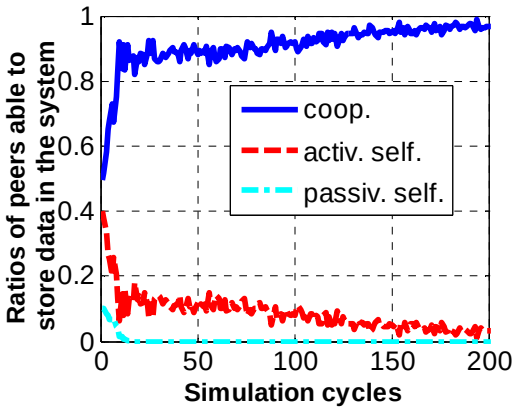

Figure 9 Ratios of cooperative and selfish peers able to store data in the system. λ=0.5, n=100, r=7, m=5, η=0.5 (0.4 active selfishness, 0.1 passive selfishness).

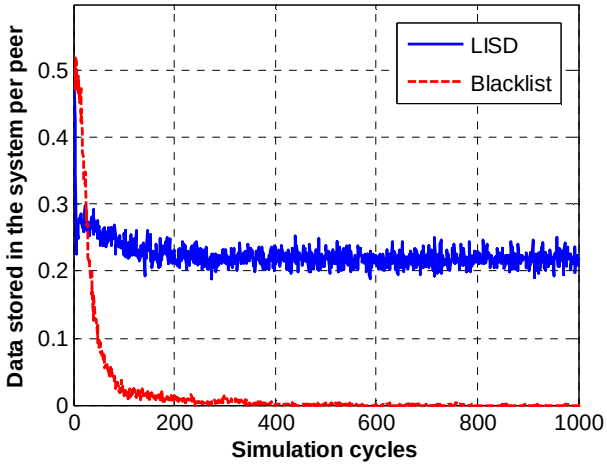

Figure 10 Data stored in the system per peer for LISD and blacklist trust models. λ=0.5, n=100, r=7, m=5, η=0.5 (0.4 active selfishness, 0.1 passive selfishness), failure rate=0.5%.

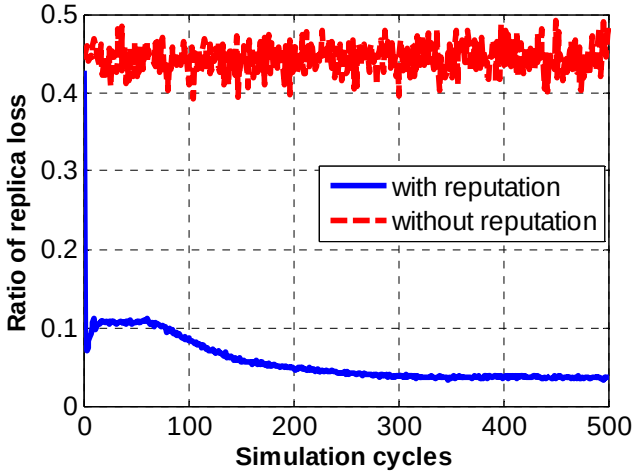

Figure 11 Average ratio of data loss using our reputation mechanism. λ=1, n=100, r=7, m=5, η=0.8 (0.7 active selfishness, 0.1 passive selfishness).

Reputation computation. Our trust model (linear increase sudden death) is compared to a blacklisting mechanism. Figure 10 depicts the amount of data stored in the system per peer and per simulation cycle, using LISD and blacklisting trust models. The figure demonstrates that with a failure rate of 0.5% the blacklisting-based model has a lot of false positives and may cause the system to collapse, i.e., no peer is able to store data in the system. However, with our trust model, selfish peers are quickly filtered out compared to the blacklisting model, and then cooperative peers if they fail may still have chances to regain the confidence of peers.

Enhancing data availability. The availability of data in a storage system is generally increased with data redundancy mechanisms (e.g., replication, erasure codes). We show in Figure 11 that data availability may still be jeopardized due to the high selfishness of some peers, and can be enhanced using our reputation mechanism. The figure depicts the ratio of data replica loss for a storage system using our reputation mechanism, and for a simple one based on a random selection of peers for storage. For the simple model, data is destroyed if the owner selects r selfish holders (with a probability η^r). The storage system with our reputation mechanism has a low ratio of data loss as a result of the eviction of selfish peers from participating in the group (as shown earlier).

6. Related work

There have been some reputation-based approaches for inciting cooperation in P2P storage systems particularly for backup applications. The following presents some reputation schemes that mostly reflect this literature.

The Free Haven project [9] consists of a set of servers called servnet community where each server hosts data from other servers in exchange of the opportunity to store data of its own in the servnet. Cooperation incentives relies on a trust module on each server that maintains a database of each other server in the servnet, logging past direct experience as well as what other servers have said. The reliability of storage is mainly based on data redundancy in the servnet. The Cooperative Internet Backup Scheme [10] proposes to enhance data reliability by allowing peers to periodically challenge their partners by requesting them to send a block of the stored data. The trust model of the scheme is based on a blacklisting mechanism: if a partner is detected of destroying data voluntarily many times beyond some threshold, the peer may decide to establish a backup contract with a different partner. The approach thwarts selfishness of storage peers by punishing them using the tit-for-tat strategy. However, these peers may still be able to store their data elsewhere in the system. Our solution is more adapted to storage applications: results of periodic storage checking are used in building a reputation mechanism that allows the filtering out of malicious peers from the storage system. Compared with the Free Haven approach, our mechanism does not require reputation information to be propagated between peers, hence preventing the damaging effect of liars in the reputation mechanism. Moreover, both [9] and [10] did not study the security of their approaches against selfish or malicious behaviors.

7. Conclusion

We described a new local reputation mechanism for P2P storage systems in which peers' observations originate from periodic verifications of data stored in the system. This approach allows a fast isolation of selfish peers, and prevents several further malicious behaviors, as illustrated by a probabilistic model and simulations based on fixed peer strategies. Additionally, we proposed a group-based design for the reputation management that may fit other types of networks such as social networks.

As future work, we plan to validate the ability of our reputation mechanism to incite peers to choose a cooperative behavior over a selfish one using for instance evolutionary game theory models.

8. References

[1] AllMydata web site: http://www.allmydata.com/

[2] Wuala web site: http://wua.la/en/home.html

[3] Ubistorage web site: http://www.ubistorage.com/

[4] Nouha Oualha, Melek Önen, and Yves Roudier, "A Security Protocol for Self-Organizing Data Storage", to appear in *IFIP Sec 2008*.

[5] 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.

[6] Patrick P. C. Lee, John C. S. Lui and David K. Y. Yau, “Distributed collaborative key agreement and authentication protocols for dynamic peer group”, *IEEE/ACM Transactions on Networking*, 2006

[7] François Lesueur, Ludovic Mé, Valérie Viet Triem Tong, “Contrôle d'accès distribué à un réseau Pair-à-Pair”, *SAR-SSI 2007*, Annecy, France.

[8] François Lesueur, Ludovic Mé, and Valérie Viet Triem Tong**, “**A Sybilproof Distributed Identity Management for P2P Networks”, *Proceedings of the 13th IEEE Symposium on Computers and Communications (ISCC) 2008*, IEEE Computer Society, Marrakech, Morocco.

[9] Roger R. Dingledine, “The Free Haven project: Design and deployment of an anonymous secure data haven”, Master’s thesis, MIT, June 2000.

[10] Mark Lillibridge, Sameh Elnikety, Andrew Birrell, and Mike Burrows, “A Cooperative Internet Backup Scheme”, In *Proceedings of the 2003 Usenix Annual Technical Conference*, pp. 29-41, San Antonio, Texas, June 2003.