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

Research Report RR-08-226

Reputation and Audits for Self-Organizing Storage

22 July 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

Reputation and Audits 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 compares reputation to an audit-based approach in which peer observations are carried out through the periodic verification of a proof of data possession, and show how the latter approach better addresses the aforementioned issues of inciting cooperation in P2P storage.

Keywords: Peer-to-peer, trust establishment, reputation, audits, distributed storage.

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 Tahoe [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.

Generally, 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 an audit-based 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 assessing peers' behavior 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 the audit-based mechanism, and presents the attacks that this system is exposed to. Section 3 compares the audit-based approach to reputation and particularly proves the satisfactory use of direct observations in estimating reputation values. Section 4 discusses implementation issues of the audit-based mechanism on top of a P2P storage system, notably regarding the mitigation of denial of service attacks on the mechanism. Section 5 evaluates the security of the proposed mechanism with respect to the attacks given in section 2. Section 6 validates the ability of the audit-based mechanism to filter out selfish peers from the storage system and to improve the availability of stored data. Section 7 covers related work. Section 8 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.
- **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 any 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 a 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 Vs. Audits

The trustworthiness 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.

A reputation-based approach for P2P storage applications generally allows estimating the trustworthiness of a given peer based on experiences and observations of its past behavior towards the actual estimator or other peers. Similarly, the audit-based approach, that we propose, relies on the estimation of the trustworthiness of this very peer based on experiences of the estimator, solely as a data owner or its observations obtained from audits of other peers' data, in the role of a verifier. The following gives an evaluation of both approaches based on an analytic model.

3.1. 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 two different cases:

- **Audits:** observations based on storage and verification 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.
- **Reputation:** observations based on peer's experiences and also recommendations: p_1 takes into account both its personal interactions and opinions expressed by other peers with respect to p_2. The reputation model is inspired from [15] where reputation computation is based on a subset of information provided by randomly chosen peers.

Audits: The probability that p_1 knows about the behavior of p_2 is equal to:

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

λ being the average storage rate of peers and n being the number of peers (the considered time unit is the time period between two verification operations).

Since personal observations are always correct, the estimated observation quality may only take two values: 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:

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

Reputation: The probability that p_1 knows about the behavior of p_2 is equal to:

$$\Pr ob[p_1\, knows\, p_2] = \theta_2 = 1 - (1 - \theta_1)^{\gamma \times n}$$

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

$$t_{k,k'} = (1-w)\overline{o} + w(k\overline{o} + k'\underline{o})/(k+k')$$
$$t'_{k,k'} = w(k\overline{o} + k'\underline{o})/(k+k')$$

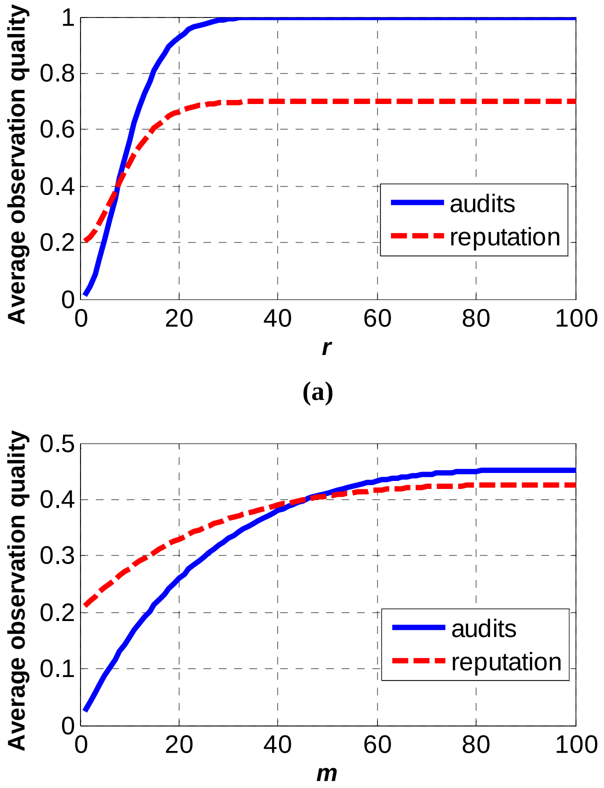
w being the weight that p_1 gives to averaged system-wide observations with respect to local observations. For $0\leq k \leq(1-\eta)\times\gamma\times n$ and $0\leq k' \leq\eta\times\gamma\times n$, we have:

$$\Pr ob[\tilde{o}_2 = t_k] = (C^k_{(1-\eta)\gamma n}\theta_1^{k+1}(1-\theta_1)^{(1-\eta)\gamma n-k})$$
$$\times(C^{k'}_{\eta\gamma n}\theta_1^{k'}(1-\theta_1)^{\eta\gamma n-k'})$$
$$\Pr ob[\tilde{o}_2 = t'_k] = (C^k_{(1-\eta)\gamma n}\theta_1^{k}(1-\theta_1)^{(1-\eta)\gamma n-k+1})$$
$$\times(C^{k'}_{\eta\gamma n}\theta_1^{k'}(1-\theta_1)^{\eta\gamma n-k'})$$
$$\Pr ob[otherwise] = 0$$

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:

$$\tilde{o}_2 = \theta_1(1-w) + w((1-\eta)\times\overline{o} + \eta\times\underline{o})$$



(a)

(b)

Figure 1 Average observation quality: (a) varying *r* and (b) varying *m*. *n*=100, *λ*=0.2, *γ*=0.3, *r*=3, *m*=5, *w*=0.5, *η*=0.3.

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

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

$$\tilde{o}_2=\theta_1(1-w)+w(1-2\eta)$$

The average quality of observations is computed in the two cases. Figure 1 shows that the best quality obtained depends very much on the replication rate. If the replication rate is low (simple data redundancy), the reputation outperforms the audit-based approach; however, if the replication rate is high (more than 10 replicas using for example erasure codes), the audit-based approach is the best way to observe. The number of verifiers has also an impact on both approaches: increasing *m* leads into an increase on the observation quality of the two approaches with a more significant increase of the audit-based approach.

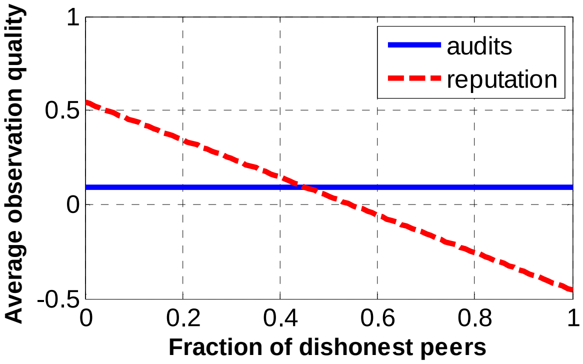
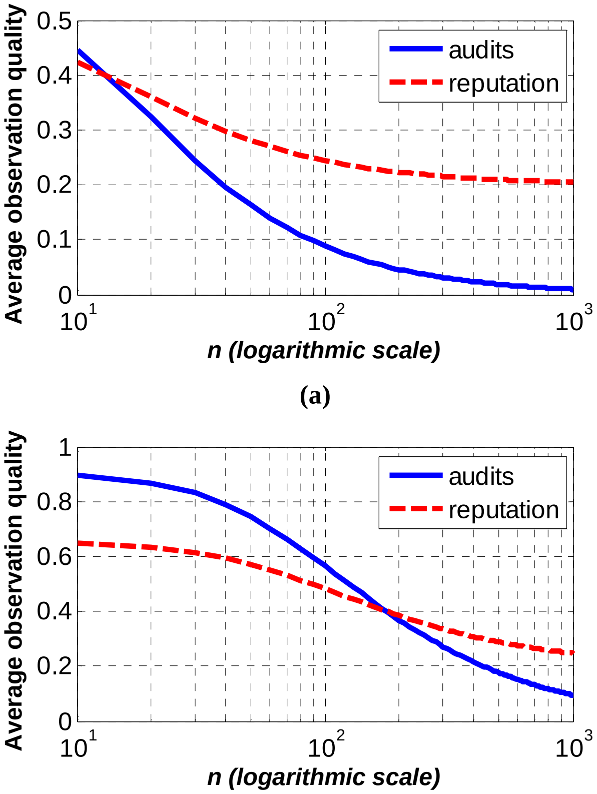


Figure 2 Average observation quality varying the fraction of malicious peers. *n*=100, *λ*=0.2, *γ*=0.3, *r*=3, *m*=5, *w*=0.5.

If the ratio of peers that send false observations increases, the quality of observation in the case with reputation linearly decreases with this ratio, however this quality is not affected in the case of audits, as it is depicted in Figure 2.

Figure 3 shows that increasing peer population *n* leads to a decrease in the quality of observations in both approaches, especially the audit-based one. Small peer populations are more in favor of audit-based approach than reputation; whereas large peer populations are more advantageous for reputation if the replication rate is small than for audit-based approach.



(a)

(b)

Figure 3 Average observation quality varying the number of peers for (a) *r*=3 and (b) *r*=10. *λ*=0.2, *γ*=0.3, *m*=5, *w*=0.5, *η*=0.3.

3.2. Summary

The study of the analytic model demonstrates that the audit-based approach for observing peer behavior outperforms reputation if the data replication rate is high (e.g., erasure coding) and with small peer population. Moreover, the approach is robust against liars, and it does not require propagation of information which avoids the problem of rumor spreading.

Since the audit-based approach works better for small population, we propose in the following section a group-based architecture for the P2P storage and a through description of its features.

4. Implementing Audits with Storage

In the P2P storage system, we rely on the construction of groups in which we evaluate peer behavior. 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.

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 it is badly audited by a growing number of group members until becoming totally isolated 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.

Group members are in a structured Distributed Hash Table (DHT) such as CAN [11], Chord [13], Pastry [12], or Tapestry [14]. A DHT consists of a number of peers having each a key Key_{Peer} in the DHT space, which is the set of all binary strings of some fixed length. We assume that the DHT provides a secure lookup service (see [16] and [17]): a peer supplies an arbitrary key (an element in the DHT space), and the lookup service returns the active node in the DHT that is the closest to the key.

In the group, peers have unique identities in the DHT. 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 audit-based P2P storage system allows peers to delegate the verification of their data to other volunteer peers, the verifiers, and also to only accept to store data of well-behaved peers.

4.2.1 Verifier selection

A data owner desiring to store a data replica in the system may randomly choose verifiers to whom it will send a verification request. The random selection of verifiers may be based on a random operation proper to the owner, for example the identity of the verifier i can be the closet key to the value $Key_{Verifier}=Hash(Key_{Owner}||nonce||i)$ where *Hash* is a pseudo-random function determined at group outset and *nonce* is a randomly chosen number protecting against a replay of the same operation ("||" means concatenation). From peers answering to this request, the owner selects m peers, 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.

4.2.2 Holder selection

To avoid collusion between the owner and the holder, selected verifiers will choose altogether the holder for the owner. Therefore, each verifier i commits to a randomly chosen DHT key k_i (commitment can be as simple hash operation of the key) and then sends this commitment to the owner. The owner sends the digest of verifiers' commitments to each verifier. Upon the receipt of the owner's message, verifiers will send their chosen random keys to the owner. The selected holder is the peer with the (numerically) closest key to the XORed sum of these random keys:

$$Key_{Holder} = k_1 \oplus k_2 \oplus \ldots \oplus k_m$$

The owner sends a digest of the messages received by verifiers containing there keys along with the identity of the chosen holder.

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

4.3. Interaction decision

Our trust model is based on whitelisting (see Figure 4) similarly to the Tit-For-Tat (TFT) strategy in BitTorent: peers that have correctly stored data they have promised to preserve are added to the whitelist of their observers (the data owner and its delegated verifiers). Whenever a peer detects that another peer has destroyed data it has promised to store, the latter will be removed from the whitelist. We also propose a "grace period" during which "no response" from the challenged holder is tolerated until the period times out, thus avoiding abusively isolating cooperative holders with transient connection. Newcomers to the system are probabilistically added to the whitelist. Newcomer acceptance probability may be computed based on the upload capacity of the peer and its whitelist size. This probabilistic process serves to bootstrap the storage system, but it also means that selfish peers changing their identities may probabilistically gain some advantage of that. Other trust models can be adopted like for example the Additive Increase Multiplicative Decrease (AIMD), the Linear Increase Sudden Death (LISD), or blacklisting mechanisms.

A peer accepts to only serve peers pertaining to its whitelist: it stores their personal data or periodically verifies their data availability in the system. However, a peer may accept to store its data at peers that do not pertain to its whitelist.

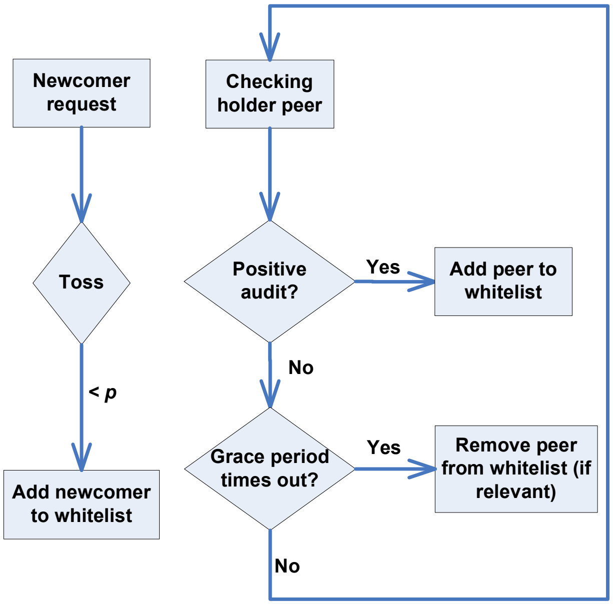


Figure 4 Whitelisting model.

5. Security Analysis

In this section, we evaluate the robustness of our proposed audit-based mechanism against the attacks exposed in 2.2.

Lying observers have no impact on our auditing mechanism since estimations are based on verification results performed by the actual estimator; thus observations are objective. Collusions between the owner and its holder or a subset of its verifiers are mitigated by the random selection of holders and verifiers. Verifiers' selection relies on a pseudo-random function and a secure routing in the DHT that can be assessed by each verifier. And, holders are randomly selected by each verifier. So, collusion between a subset of participants is prevented.

The group-based architecture of the P2P storage permits controlling peers who are joining the storage system in order to mitigate Sybil attackers. This latter may still be able to take profit of peers that are probabilistically adding newcomers to their whitelist, still this probability can be adjustable depending on peer's confidence on the system. The architecture allows also a rapid knowledge about the behavior of group members based on audits, and then peers are able to refuse storage to non cooperating peers, hence limiting free-riders.

6. Simulation Experiments

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

6.1. Framework

The self-organizing network is a modeled as a closed set of peers with a fixed storage rate and several behavior strategies. We consider the following strategies:

- **Cooperation** whereby the peer concedes storage space for other peers' data and sends correct verification results to owner.
- **Free riding** whereby the peer free rides by using the storage offered in the network without contributing its equal share. We distinguish between:
    - **rational** peers that change their strategies to cooperation if they cannot store data in the system; and whenever they are able to store again they return to their original strategy; whereas,
    - **irrational** peers persist in free-riding.
- **Active selfishness** whereby the peer only probabilistically conserves data stored and gives incorrect verification results to the owner with some probability. We distinguish between rational and irrational actively selfish peers:
    - **rational** peers will change their strategy if they cannot anymore store data in the system; and whenever they are able again to do that they return to selfishness; whereas,
    - **irrational** peers will keep their selfish strategy.

6.2. 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 only cooperative peers are the active consumers of the storage in the system.

Exclusion of selfish owners. Figure 5 demonstrates that selfish peers have less capability over time to store data in the system; however, cooperative peers are becoming the majority of data owners in the storage system. Free-riders are excluded from storing data in the system before active selfish peers, because the latter cooperate at first by storing data then they destroy them which may slow their detection.

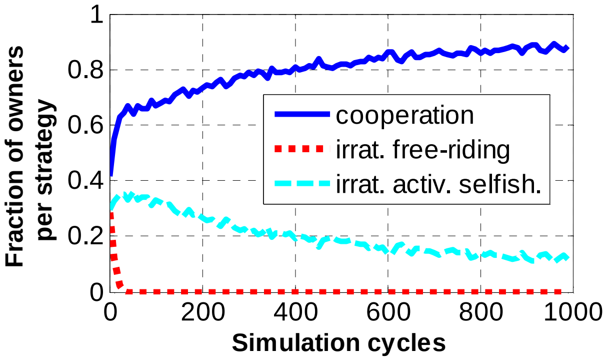


Figure 5 Ratio of owners per strategy. ***n*=100, *λ*=0.3, *r*=3, *m*=5, *p*=0.01, p_{in}=0.3, p_{out}=0.02, 0.4% cooperators, 0.3% irrational free-riders, 0.3% irrational actively selfish peers.**

Inciting cooperation. Figure 6 shows the decreasing of the fraction of rational free-riders and rational actively selfish peers over time. This means that with our audit-based mechanism, peers are motivated to replace their selfish strategy by cooperation strategy.

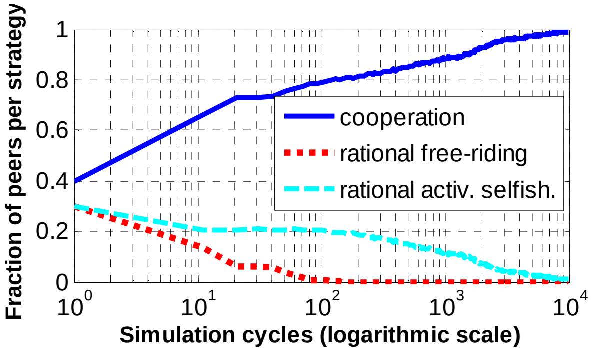


Figure 6 Fraction of peers per strategy. *n*=100, *λ*=0.3, *r*=3, *m*=5, *p*=0.01, p_{in}=0.3, p_{out}=0.02, 0.4 cooperators, 0.3 rational free-riders, 0.3% rational actively selfish peers.

Reputation vs. audits. Figure 7 depicts the evolution of cooperative owners over time using reputation and audits and for different initial fraction of cooperators in the system. In the case of reputation, actively selfish peers always give false observations to the requester; however, free-riders never give any observation. The figure demonstrates that the audit-based mechanism outperforms reputation for a system with high active selfishness. When the fraction of active selfishness is low, reputation slightly improves on audits at the beginning, and then it is surpassed by the audit-based approach.

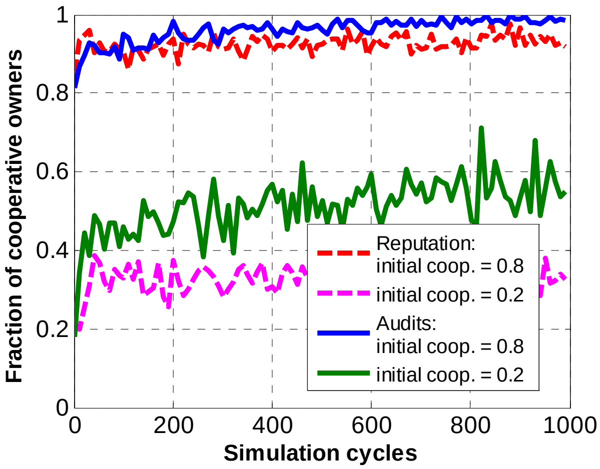


Figure 7 Fraction of cooperative owners for reputation and audits varying the initial fraction of cooperators. *n*=100, *λ*=0.3, *r*=3, *m*=5, *p*=0.01.

Newcomer's acceptance. Figure 8 shows that a large probability *p* for newcomers' acceptance slows the convergence time of the system to a system free from selfish owners. This slowing down becomes less significant with large probability *p*.

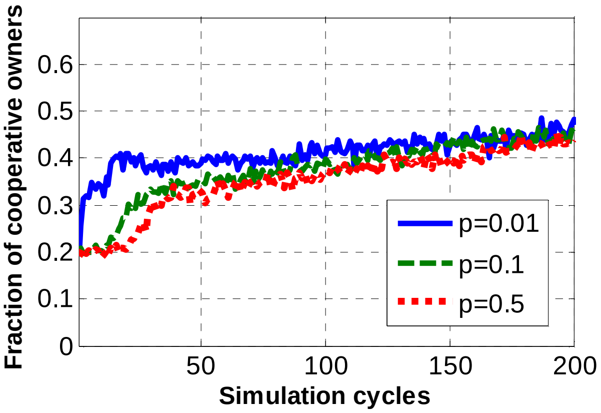


Figure 8 Fraction of cooperative owners varying the probability of newcomer's acceptance *p*. *n*=100, *λ*=0.3, *r*=3, *m*=5, p_{in}=0.3, p_{out}=0.02, ¼ cooperators, ¼ free-riders, ¼ rational and ¼ irrational active selfishness.

Data reliability. The reliability of data in a storage system is generally increased with data redundancy mechanisms (e.g., replication, erasure codes), as illustrated in Figure 9 with very low data loss. However, the figure shows also that the amount of data injected into the storage system is lower than the storage rate. This is due to several factors. First of all, there is the probability of acceptance *p* that slows the bootstrap of the storage system. Then, there is the gradual exclusion of selfish peers that limits the number of peers able to store data in the system (which explains the small peak at 25 simulation cycles). An finally, there is the churnout of the P2P system by which some cooperative peers are removed from the whitelist because they were offline for a period higher than the grace period (selfishness detection with false positives).

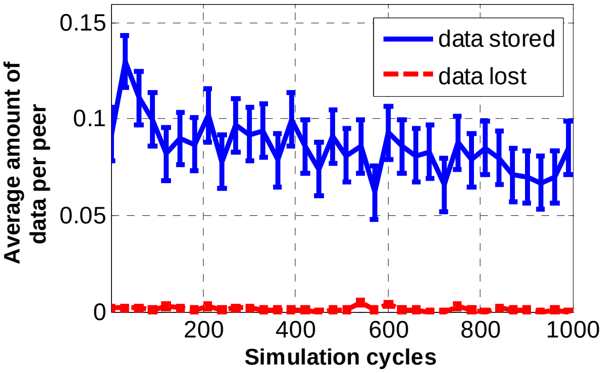


Figure 9 Average amount of data stored and lost per peer. *n*=100, *λ*=0.3, *r*=3, *m*=5, *p*=0.01, p_{in}=0.3, p_{out}=0.02, ¼ cooperators, ¼ free-riders, ¼ rational and ¼ irrational active selfishness.

7. 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.

8. Conclusion

We described an audit-based mechanism for P2P storage systems in which peers' observations originate from periodic verifications of data stored in the system. We demonstrated that the audit-based approach is more robust to selfish behavior than reputation, which may be a better choice for today's commercial storage systems where peers have economic compensation for storing data, and thus they may be motivated to give false recommendations in order to gain fame. Additionally, we proposed a group-based design for audits management that may fit several types of networks such as social networks.

9. 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 N° 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.

[11] Sylvia Ratnasamy, Paul Francis, Mark Handley, Richard Karp, and Scott Shenker, “A scalable content-addressable network”, In *Proceedings of SIGCOMM*, San Diego, CA, Aug. 27–31, 2001.

[12] Antony Rowstron and Peter Druschel, “Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems”, In *IFIP/ACMInternational Conference on Distributed Systems Platforms*, Heidelberg, Germany, Nov. 2001.

[13] Ion Stoica, Robert Morris, David Karger, M. Frans Kaashoek and Hari Balakrishnan, “Chord: A scalable peer-to-peer lookup service for internet applications”, In *Proceedings of SIGCOMM*, San Diego, CA, Aug. 27–31, 2001.

[14] Ben Y. Zhao, John Kubiatowicz, and Anthony D. Joseph, “Tapestry: An infrastructure for fault-tolerant wide-area location and routing”, Technical Report UCB//CSD-01-1141, University of California, Berkeley, Apr. 2000

[15] Emmanuelle Anceaume and Aina Ravoaja, “Incentive-Based Robust Reputation Mechanism for P2P Services”, Research Report PI 1816 (2006), IRISA, http://hal.inria.fr/inria-00121609/fr/

[16] Emil Sit and Robert Morris, “Security Considerations for P2P Distributed Hash Tables”, IPTPS 2002.

[17] Miguel Castro, Peter Druschel, Ayalvadi Ganesh, Antony Rowstron and Dan S. Wallach, “Secure routing for structured peer-to-peer overlay networks”, *Symposium on Operating Systems and Implementation*, OSDI’02, Boston, MA, December 2002.