Reputation and Audits for Self-Organizing Storage

Nouha Oualha
EURECOM, Sophia Antipolis, France
nouha.oualha@eurecom.fr

Yves Roudier
EURECOM, Sophia Antipolis, France
yves.roudier@eurecom.fr

ABSTRACT
Reputation systems have demonstrated their interest in stimulating cooperation in peer-to-peer (P2P) systems. Their key operation relies on collecting, processing, and disseminating the feedback about some peers' past behavior in order to boost their cooperation, albeit this is susceptible to collusion and bashing. Additionally, estimating 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 where peer observations are carried out through the periodic verification of a proof of data possession, and shows how the latter approach better addresses the aforementioned issues of inciting cooperation in P2P storage.

Categories and Subject Descriptors
C2.4 Distributed Systems.

General Terms
Reliability, Security.

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

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 [3], and Ubistorage [4] 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 in the meantime 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. Typically, approaches for building reputation systems presume that peers engage in repeated interactions, and that the information of their past doings that is taken as their reputation is indicative of their future performance. Still, they are also making simplifying assumptions on the instantaneous propagation around the system of the indirect reputation information in question and on the willingness of peers to correctly and fairly propagate such information. We propose in this paper to use remote storage auditing for observation thereby serving a twofold objective: inciting peers to check the availability of others' data and at the same time assessing peers' behavior using the very results of verification.

Commercial storage systems such as AllMyData [1] and Ubistorage [4] do not have any cooperation incentive mechanisms, they assume that data which have been accepted by a storage server will be "retained and retrieveable until the lease is cancelled or expires, or until the server fails" [2]. The exception is made with Wuala [3] that uses a reputation-based approach to motivate peers to stay online. In the research community, there have been several works on reputation for P2P storage systems (e.g., [10], [11]); but they did not evaluate the security aspects of their approaches against selfish or malicious behaviors. In this paper, we examine the security of our solution in terms not only of attack mitigation but also in terms of the quality of reputation information used and the process of such information to identify and subsequently punish non cooperative peers in the storage system.

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 auditing, and presents the attacks that this system is exposed to. Section 3 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 4 compares the audit-based approach to reputation and particularly proves the satisfactory use of direct observations in estimating reputation values that are used in isolating selfish peers. Section 5 finally presents our concluding remarks.

2. P2P STORAGE: AN OVERVIEW
A P2P storage application allows data owner peers to store their personal data in replicas at several data holder peers. A stored data replica is periodically checked by verifier peers on behalf of the owner. 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.

We review the actors of the system: the *owner* that stores its data to a set of *r holders* which keep the data until their retrieval by the owner. In addition, each holder is monitored by a set of *m verifiers* (the owner may participate in the verification process).

2.1 Data possession verification

The verification process relies on a secure data possession verification protocol. As discussed in [5] and [6], the protocol assumes that the verifier possesses metadata information allowing it to properly check data storage at the holder. The verification is based on challenge-response messages exchanged between the verifier and the holder (see Figure 1). The verifier constructs a time-variant challenge message and sends it to the holder. The holder derives the response from the received challenge and the data that it is storing, and then returns the response to the verifier. Upon reception of the holder's response, the verifier checks whether the response is valid using the verification metadata for deciding if the holder is still storing the data.

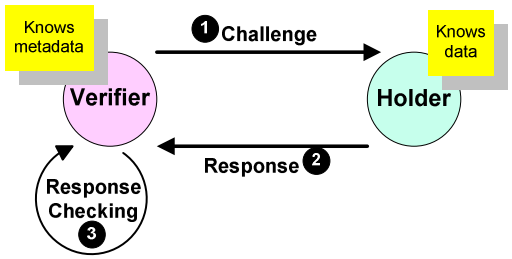


Figure 1 Verification of data possession

No response from the holder cannot be taken as an undeniable indicator of deliberate data destruction by the holder, because it may result from connection break between the verifier and the holder, an offline holder, or even failure of this latter. However, persistent no response indicates that the holder has destroyed the data it promised to store, and thus it is considered by the verifier as selfish.

The verification process is periodically initiated by the verifier. For example, consider a verifier performing verification after each time period T and a holder appointed to this verifier destroying them after a time τ of their storage. Data destruction will be detected by the verifier with a delay that equals $\lceil \tau/T \rceil \times T - \tau$. So, the frequency of these data verifications tunes the granularity of holder's behavior evaluations: the smaller the verification period T, the faster the detection of the holder's non cooperation. In a self-organizing environment like the P2P network, verifiers and holders may be offline; therefore, we suggest that the data verification protocol may be rather opportunistically executed by verifiers, without exceeding the limits of a large time period interval.

2.2 Data storage

Storing data in a P2P system is composed of several phases:

- **Verification delegation:** The owner delegates the task of verifying data stored in the system to well reputed peers. It sends them metadata information (containing e.g., delegation certificate, data digest, etc) that allows such verification.
- **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. 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.

2.3 Reputation & Audits

We estimates the trustworthiness of a peer based on the observation of its past behavior. The semantics of the collected information 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 observations.

Reputation. A reputation-based approach for P2P storage applications allows estimating the trustworthiness of a given peer based on experiences and observations of its past behavior towards the actual estimator or other peers. This means that owners and verifiers will disseminate in the storage system their personal observations about the holders they had interacted with. Peers will collect these observations about a given peer to decide whether to store or verify data of the very peer.

Audits. The audit-based approach, which we propose, relies on the estimation of the trustworthiness of a given peer solely based on personal experiences of the estimator. The estimator will use its observation, as a data owner or its observations obtained from audits of other peers' data, in the role of a verifier. Again, the observed peer is the holder. We believe that the periodic verification of data will improve the accuracy of such estimations.

2.4 Adversary model

In our model, we consider the adversaries that do not correctly follow the roles as peers (owner, data holder, or 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 they 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 storage community, or that may destroy some data they promised to keep in order to optimize their own storage resources. Such peers never play the role of holder or verifier of other peers' data.

- **Collusion between holders:** Holders collude so that only one of them keep the 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 [5] and [6]. However, this is obtained with some cost, because personalization generally means that metadata information allowing a given verifier to check storage at a holder is also personalized. A verifier checking the same data at different holders must then hold more information consisting of different metadata corresponding to the personalized data replicas.
- **Maliciousness:** Malicious peers aim to destroy the 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 & audits 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 to increase 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. One way to mitigate this attack is to have the verifiers altogether select the holder on behalf of the owner, thus guaranteeing to verifiers that the owner and the holder do not know each other a priori.
- **Collusion between holder and verifier:** The aim of such a 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. IMPLEMENTING AUDITS WITH STORAGE

This section aims at proving the feasibility of the reputation-based and the audit-based approaches for P2P storage applications. In the 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.

3.1 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 or reputed 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 initially constructed by an authority like partnership in [11] 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., [7], [8]). The group key controls the access to the group, and ensures secure and private communication between its members.

Group members are organized in a structured Distributed Hash Table (DHT) such as CAN [12], Chord [14], Pastry [13], or Tapestry [15]. 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 [17] and [18]): 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.

Peers, in the group, 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 [9].

3.2 Self-organizing peer selection

In the P2P storage system, peers are able 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.

3.2.1 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.

3.2.2 Holder selection

To avoid collusion between the owner and the holder, the 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 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 their keys along with the identity of the chosen holder.

It is clear that the process of selecting holders 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.

3.3 Interaction decision

Our trust model is based on whitelisting (see Figure 2) similarly to the Tit-For-Tat (TFT) strategy in BitTorrent [19]: peers that have correctly stored data they have promised to preserve are added to the whitelist of their observers (observers may be the owner and its delegated verifiers or the peers to which this observation was propagated in the reputation case). 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.

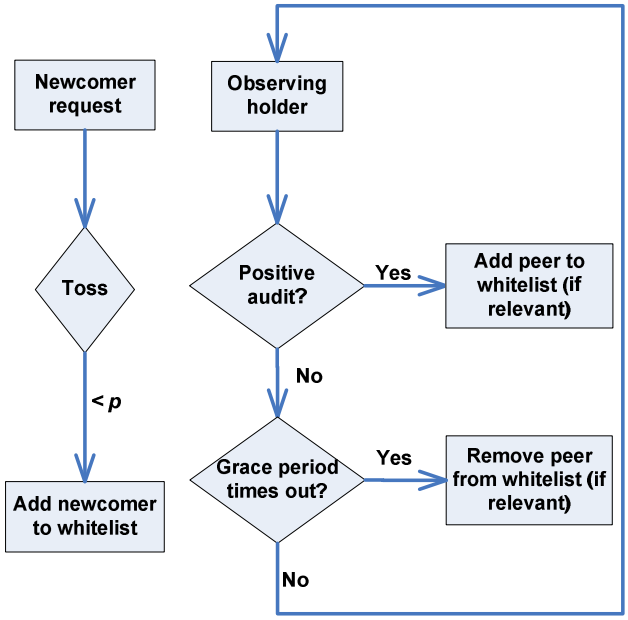


Figure 2 Whitelisting model.

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.

4. REPUTATION VS. AUDITS

In this section, we examine two questions relevant to understanding how efficient reputation and audits are in 1) collecting observations and 2) processing them. We propose an analytic model to study the quality of observations obtained in both approaches and a simulation based-experimentation to evaluate their actual process of selfishness detection in the P2P storage system.

4.1 Analytic model

This section discusses the advantageous of choosing a reputation or an audit-based approach over the other with respect to the level of correctness of gathered observations.

4.1.1 Model description

Considering two peers p_1 and p_2, where p_1 desires to have correct observations on p_2. Peer p_1 may make a correct observation itself or may receive observations from other peers in the system that may be correct or incorrect. Let η denotes the fraction of incorrect indirect observations that may be obtained from the system. These incorrect observations are conveyed by selfish or malicious peers (this type of peers may also send correct observations, but it is assumed that they always send incorrect observations). Cooperative peers transmit correct observations.

Table 1 Notation used

Symbol	Description
p_1	The estimator desiring to have correct observations about a given peer
p_2	The observed peer
r	Number of holders (number of stored data replicas)
m	Number of verifiers per one holder
n	Number of participants (peers) to the group
η	Fraction of incorrect indirect observation from peers
$\bar{o}$	Maximum observation quality level
$\underline{o}$	Minimum observation quality level
$\tilde{o}$	Average quality level of the estimated observation by p_1
λ	Average storage rate of peers
γ	Fraction of peer population to which the reputation is propagated
θ_1	The probability that p_1 has an observation of p_2 in the audit-based approach
θ_2	The probability that p_1 has an observation of p_2 in the reputation-based approach

We define a quality level for the estimated observation with two extrema: $\bar{o}$ and $\underline{o}$. An observation of quality $\bar{o}$ 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 average estimated observation quality of p_1, denoted $\tilde{o}$, is then derived for two cases: reputation and audit-based approaches. This average $\tilde{o}$ indicates the level of correctness of the estimated observation obtained by p_1: the more $\tilde{o}$ approximates $\bar{o}$, the more the estimated observation in average is correct; whereas, the more it approximates $\underline{o}$, the more the observation is incorrect. For an average $\tilde{o}$ that equals $(\bar{o} + \underline{o})/2$,

we cannot claim that the observation is correct or incorrect (e.g., case p_1 has no observations about p_2).

The average $\tilde{o}$ is computed 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 [16] where reputation computation is based on a subset of information provided by randomly chosen peers.

Table 1 summarizes the notation used in the proposed model.

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

$$Prob[p_1\ knows\ p_2] = \theta_1 = 1 - \left(1 - \frac{\lambda r}{(n-1)}\right) \times \left(1 - \frac{\lambda r}{(n-1)} + \frac{\lambda r}{(n-1)}\left(1 - \frac{m}{(n-2)}\right)^r\right)^{n-2} \quad (1)$$

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

The probability (1) takes into account the probability that p_1 chooses p_2 as a holder of its data (p_1 stores data at rate λ) and the probability that another peer from the n-2 remaining peers chooses p_2 as a holder and p_1 as a verifier for it.

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

$$Prob[\tilde{o}_1 = \bar{o}] = \theta_1$$
$$Prob[\tilde{o}_1 = \underline{o}] = 0$$
$$Prob[\tilde{o}_1 = 0] = 1 - \theta_1$$

On average, we have:

$$\tilde{o}_1 = \theta_1 \times \overline{o} \quad (2)$$

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

$$t_{k,k'} = (1-w)\overline{o} + w\frac{(k\overline{o}+k'\underline{o})}{k+k'} \quad (3)$$

$$t'_{k,k'} = w\frac{(k\overline{o}+k'\underline{o})}{k+k'} \quad (4)$$

γ being the fraction of the peer population to which the reputation is propagated, and w the weight that p_1 gives to averaged system-wide observations with respect to local observations.

(3) gives the average observation quality taking into account correct observations obtained from the owner itself and cooperative peers, and incorrect observations obtained from selfish or malicious peers. (4) only considers indirect observations.

For $0 \le k \le (1\text{-}\eta)\times\gamma\times n$ and $0 \le k' \le \eta\times\gamma\times n$, we have:

$$Prob[\tilde{o}_2 = t_{k,k'}] = \theta_1 (C^k_{(1-\eta)\gamma n} {\theta_1}^k (1-\theta_1)^{(1-\eta)\gamma n-k}) \times (C^{k'}_{\eta\gamma n} {\theta_1}^{k'} (1-\theta_1)^{\eta\gamma n-k'}) \quad (5)$$

$$Prob[\tilde{o}_2 = t'_{k,k'}] = (1-\theta_1)(C^k_{(1-\eta)\gamma n} {\theta_1}^k (1-\theta_1)^{(1-\eta)\gamma n-k}) \times (C^{k'}_{\eta\gamma n} {\theta_1}^{k'} (1-\theta_1)^{\eta\gamma n-k'}) \quad (6)$$

The value $C^k_{(1-\eta)\gamma n}$ (respectively $C^{k'}_{\eta\gamma n}$) is the number of combinations of k (respectively k') peers from the set of cooperative (respectively non cooperative) peers from which p_1 gathers observations.

(5) consists of the probability that p_1 interacted with p_2, the probability that k peers from the set of $(1\text{-}\eta)\times\gamma\times n$ cooperative peers interacted with p_2 and the rest of the set did not, and also the probability that k' peers from the set of $\eta\times\gamma\times n$ non cooperative peers interacted with p_2 and the remainder of the set did not. (6) is similar with (5) but having instead the probability that p_1 did not interacted with p_2.

A certain probability of interaction is attached to the observations of both cooperative and non cooperative peers. This is due to the fact that peers have to provide cryptographic proofs (e.g., signature) that they had interactions with p_2. 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.

From (3, 4, 5, 6), the average is derived as:

$$\tilde{o}_2 = \sum_{k=0}^{(1-\eta)\gamma n} \sum_{k'=0}^{\eta\gamma n} Prob[\tilde{o}_2 = t_{k,k'}] \times t_{k,k'} + Prob[\tilde{o}_2 = t'_{k,k'}] \times t'_{k,k'} \quad (7)$$

Using the Vandermonde's identity over k and k', (7) becomes:

$$\tilde{o}_2 = \theta_1(1-w) + w((1-\eta)\times\overline{o} + \eta\times\underline{o}) \quad (8)$$

We notice that the fraction γ does not appear in (8); this is because the probability of correct observation is dependent on η that is taken as fraction and hence is not determined by the quantity of observations collected.

4.1.2 Analytic comparison

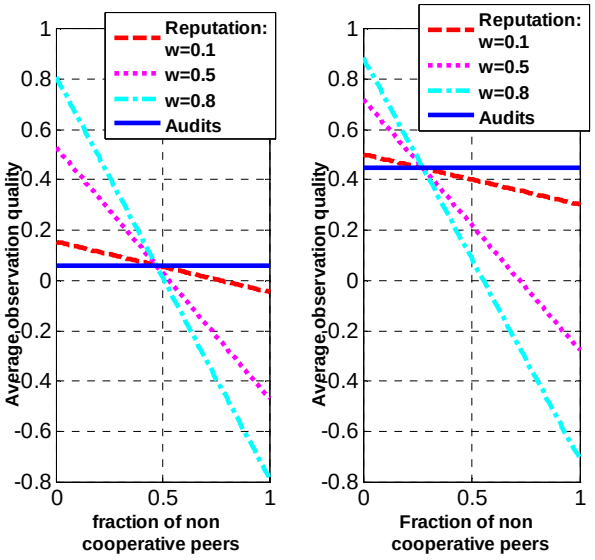
Seeking for simplicity, we choose quality observations such as: $\bar{o} = 1$ and $\underline{o} = -1$. Thus, (2) and (8) become:

$$\tilde{o}_1 = \theta_1 \quad (9)$$

$$\tilde{o}_2 = \theta_1(1-w) + w(1-2\eta) \quad (10)$$

The average quality of observations is computed for reputation and audit-based approaches in different setting. We distinguish between simple data redundancy and erasure coded data. Erasure-codes have been used in Wuala [3], AllMyData Tahoe [2] (Tahoe is free software sponsored by AllMyData), and the backup application of [11]. The usual used redundancy factor is around 3 (3 of 10 chunks are sufficient to recover the whole file in AllMyData Tahoe, i.e., replication factor=10/3~3). Therefore, we will consider two replication values r=3 and r=10 for respectively simple data redundancy and erasure coding. The size of the peer population is generally determined by the type of the network. The size of "swarms" in BitTorrent [19] ranges from 300 to 2000 peers depending on file popularity. Whereas, in social network, Dunbar's [20] rule of 150 is generally employed. Thus, we will consider two different values for peer group size: n=150 and n=2000.

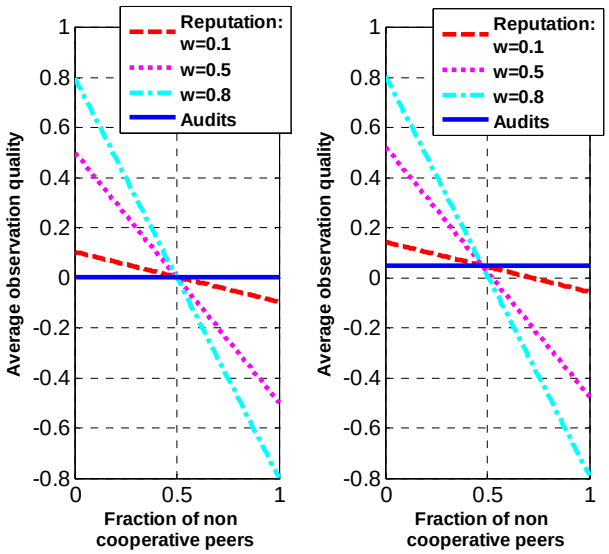
False observations. The fraction of non cooperation η has no impact on the audit-based approach. For high fraction of non cooperation in the storage system, reputation has a poorer observation quality than audits. The point on η axis at which average observation qualities of both approaches are equal varies with r and n. For small group size (n=150), the point is lower for higher replication rate. In Figure 5, this point approximates η=0.5 for r=3 and η=0.25 for r=10. This means that increasing the number of replicas r increases the performance of the audit-based approach over reputation.



(a) r=3, n=150 **(b) r=10, n=150**

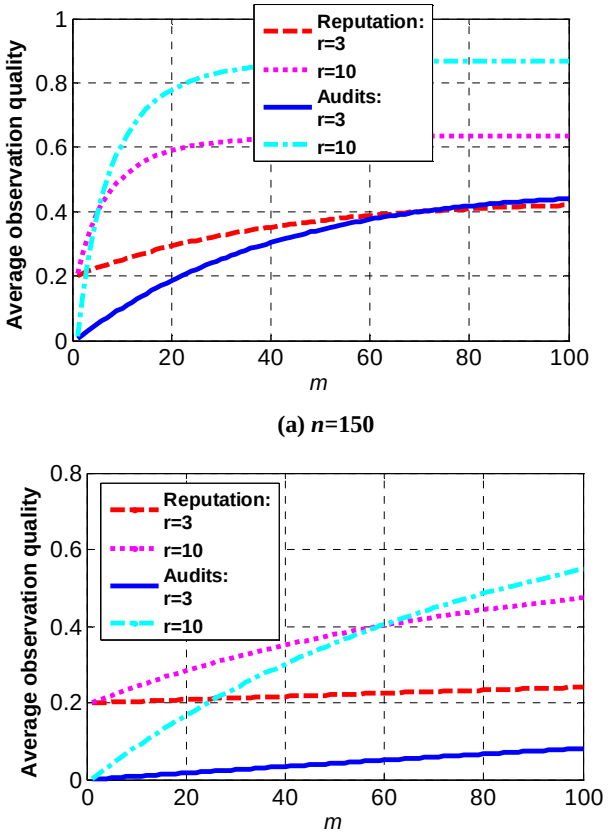
Figure 3 Average observation quality for small peer groups n=150 varying r: (a) r=3 (b) r=10. λ=0.2, m=5.

For large group size (n=2000), the point practically does not change with r=3 and r=10 (η~0.5 in Figure 4). Thus, the number of replicas has no significant impact on performance differentiation between the two approaches for large group size.



(a) r=3, n=2000 **(b) r=10, n=2000**

Figure 4 Average observation quality for large peer groups n=2000 varying r: (a) r=3 (b) r=10. λ=0.2, m=5.

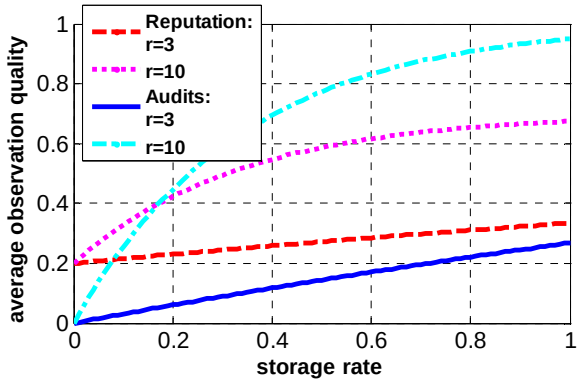


(a) n=150

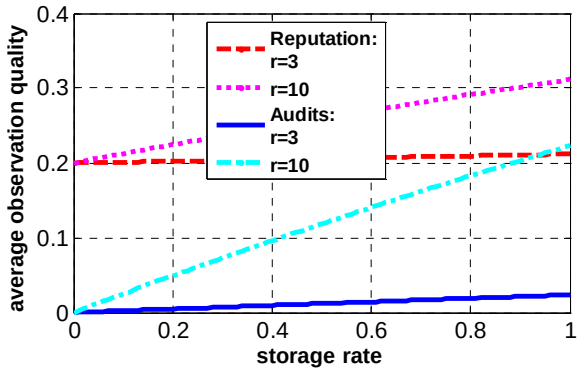
(b) n=2000

Figure 5 Average observation quality varying r and m with (a) n=150 and (b) n=2000. λ=0.2.

Number of verifiers. The number of verifiers m increases the quality of observation of the two approaches. This increase is more important for the audit-based approach than for reputation, that's why, the audit-based approach beats reputation for high value of m. The point of switch on m axis at which the observation quality of the audit-based approach outpaces reputation varies with n and r. Figure 5 shows that the point equals to m=5 for n=150 and r=10 and is higher than 100 for n=2000 and r=3.



(a) n=150



(b) n=2000

Figure 6 Average observation quality varying r and λ with (a) n=150 and (b) n=2000. m=5.

Storage rate. Increasing the storage rate λ makes the quality of observation increase for the two approaches, but more significantly for audits than reputation. Figure 6 shows that reputation is usually outperforming the audit-based approach, except the case of r=3 and n=150 where the audit-based approach is more advantageous for λ>0.2.

Summary. The study of the analytic model demonstrates that the audit-based approach for observing peer behavior outperforms reputation if the number of data replicas is high (e.g., erasure coding) and with small group peer population, as it may be the case for a social network. However, with small population, the number of peers volunteering for verification will be small, and thus using reputation may be more advantageous. Additionally, the analytic model reveals that increasing the number of interactions between peers, e.g., increasing λ, r or m, has a much better impact on the quality of observations collected by the audit based approach than by reputation. So, for an actively in demand storage system, audits are more competitive than reputation; on the contrary for a system that does work at low capacity, reputation becomes more valuable.

These results suggest that the decision to choose one approach over the other must be made by the peer itself: with the observations it has and system metrics it estimates (e.g., λ, and η), the peer can determine if it requires reputation or an audit-based approach and can as well properly establish their parameters (e.g., w, p).

4.2 Simulation experiments

This section evaluates reputation and audits in terms of selfishness detection with simulation. We implemented a custom simulator whose framework is at first described, and then results of simulation are presented and analyzed.

4.2.1 Framework

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

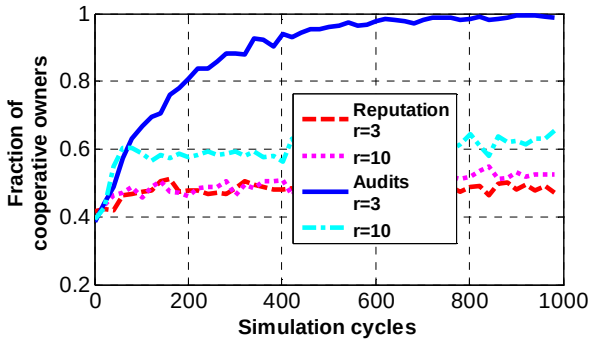
- **Cooperation** whereby the peer concedes storage space for other peers' data and sends correct verification results to owner.
- **Free riding** whereby the peer free rides by using the storage offered in the network without contributing its equal share. In a reputation-base approach, free-riders never give any observation. 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 verifiers other peers' data with some probability. In a reputation approach, actively selfish peers always give false observations to the requester. 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.

4.2.2 Simulation results

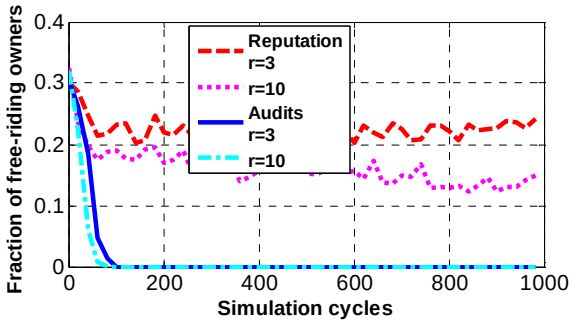
Different scenarios within the framework are simulated in order to analyze the impact of system parameters and choices on the convergence time of the storage system to a stable state where only cooperative peers are the active consumers of the storage in the system. Framework simulations are cyclic. A simulation cycle corresponds to a time period between two successive verifications.

The same system parameters as in the previous Section 4.1 are considered. Because it is prohibitive to simulate a huge group of peers, we will limit simulations to groups of size n=150. The size of the whitelist in average equals to 4 (similarly to the default "active set" size in BitTorrent [19]). So the probabilistic peer acceptance p=0.03 (~4/n). The frequency of verifications is set to every one hour (we choose to use a high frequency to accelerate the results of studied approaches). Peers may connect or disconnect from the storage system with some given probabilities, respectively denoted p_{in} and p_{out}. Generally, peers are continuously connected in average (e.g., Wuala [3]) for more than 4 hours per day (p_{in}/p_{out}>4). Hence, they are able to perform more than one verification operation per day. Finally, we suppose that

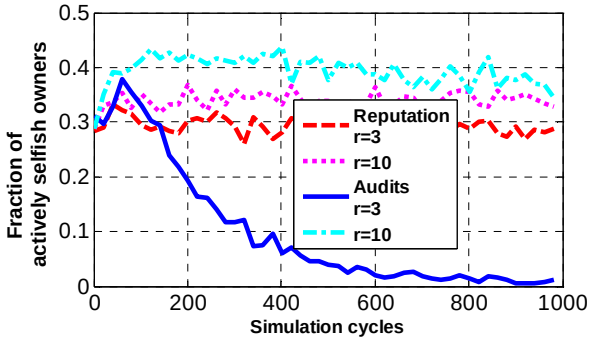
in average 30% of peers connect at the bootstrap (p_{in}=0.3, p_{out}=0.075).



(a) fraction of cooperative peers able to store data



(b) fraction of free-riders able to store data



(c) fraction of actively selfish peers able to store data

Figure 7 Average ratio of owners per strategy (a) cooperation, (b) free-riding, and (c) active selfishness. n=150, λ=0.2, m=5, p=0.03, p_{in}=0.3, p_{out}=0.075, w=0.8, initial composition: 0.4% cooperators, 0.3% irrational free-riders, 0.3% irrational actively selfish peers.

Exclusion of non cooperative owners. Figure 7 depicts the fraction of peers able to store data in the system with respect to their strategies. The figure demonstrates that reputation and the audit-based approach are able to detect and prevent non cooperative peers from utilizing storage at the system; but each approach processes this at a different pace. The figure proves that the audit-based approach is faster than reputation (with w=0.8) in excluding free-riders and actively selfish peers from storing data in the system (reputation with small w produces practically similar results as audits). Free-riding owners are first rejected before actively-selfish owners; because the latter cooperate at first by storing data before destroying them which slows their detection. This explains also the small peak at about 50 simulation cycles: the number of actively selfish owners does not increase, but in fraction it does due to the elimination of free-riding owners. We notice also that actively-selfish peers are difficult to eliminate from the set of owners if the replication rate r is high; on the contrary free-riders are quickly eliminated with high replication.

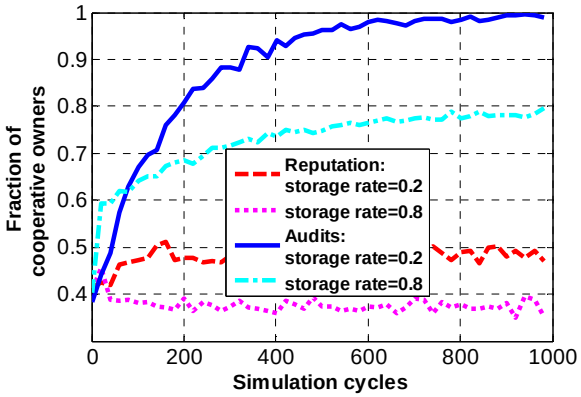
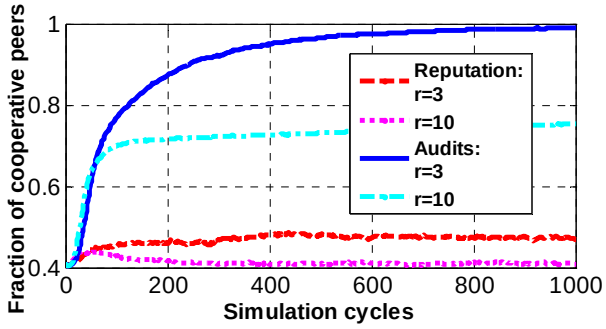


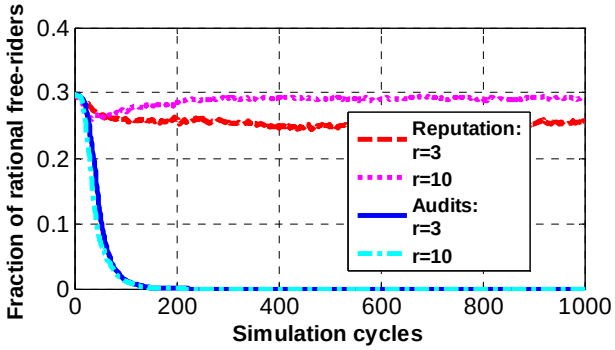
Figure 8 Average ratio of cooperative owners varying the storage rate λ. n=150, r=3, m=5, p=0.03, p_{in}=0.3, p_{out}=0.075, w=0.8, initial composition: 0.4% cooperators, 0.3% irrational free-riders, 0.3% irrational actively selfish peers.

Storage rate. The load of the storage system impacts selfish peers' detection. Figure 8 depicts the variation of the fraction of cooperative owners in the system over time, for two different storage rates: λ=0.2 and λ=0.8. The figure shows that it takes more time to make cooperators the only peers able to store data in the system with the high storage rate than with the low one. This result is relevant for both approaches. There is a rapid increase of cooperators' ratio around 50 simulation cycles (for curves with λ=0.8) due to a more efficient detection of free-riders with the high storage rate. So, high storage rate is more effective for detecting free-riders than actively selfish peers. High storage rate produces more chances for actively selfish peers to go unnoticed by accepting to store a lot of data, without eventually fulfilling their promise.

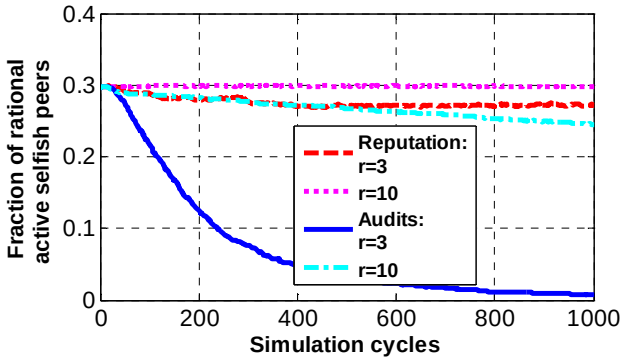
Inciting cooperation. Figure 9 depicts the fraction of rational peers in the system over time. The figure shows that cooperative behavior is becoming the most rationally advantageous strategy over time for the audit-based approach: the other strategies (free-riding and selfishness) are decreasing in population. Reputation is inciting peers to choose cooperation over selfishness for small replication rate r. For high replication value, reputation is not able to cope with false observations disseminated by actively selfish peers. So, free-riders and selfish peers are still able to store data in the system. The population of cooperators does not change a lot over time, and the populations of free-riders and selfish peers survive. The replication rate has also an impact on the audit-based approach. This impact concerns only actively selfish peers: with high replication, it is more difficult to convince rational actively selfish peers to change strategy to cooperation. This is because they have a lot of opportunities to be selected as holders so that they can temporarily counterbalance their past selfish behavior.



(a) fraction of cooperative peers in the system



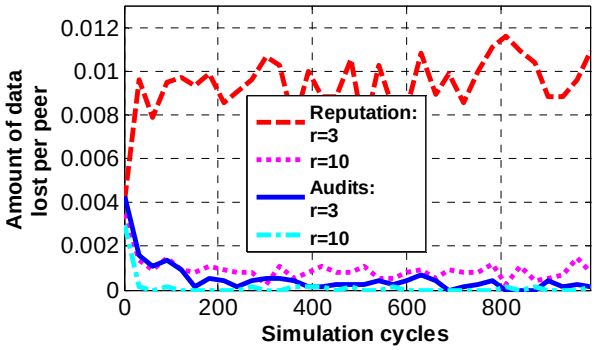
(b) fraction of rational free-riders in the system



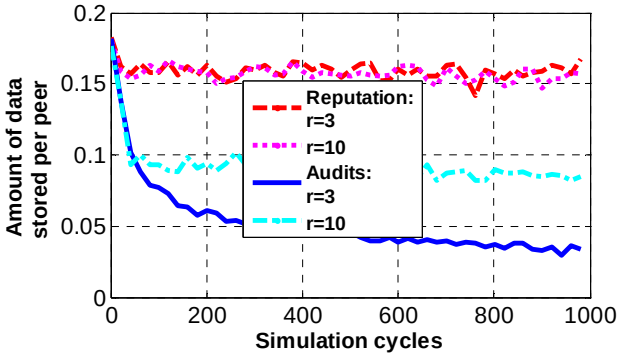
(c) fraction of rational actively selfish peers in the system

Figure 9 Average ratio of peers per strategy (a) cooperation, (b) free-riding, and (c) active selfishness. n=150, λ=0.2, m=5, p=0.03, p_{in}=0.3, p_{out}=0.075, w=0.8, initial composition: 0.4% cooperators, 0.3% rational free-riders, 0.3% rational actively selfish peers.

Data reliability. The reliability of data in a storage system is generally increased with data replication, as illustrated in Figure 10 with very low data loss. For the same low replication rate r=3, the data loss for reputation is higher than the one for audits. The figure shows that the data loss for audits decreases with time, due to peers changing their strategies from selfishness to cooperation. From the figure, we notice also that the amount of data injected into the storage system is lower than the storage rate ($<\lambda$=0.2). 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. And 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).



(a) data lost per peer



(b) data stored per peer

Figure 10 Average amount of data stored and lost per peer and per simulation cycle. n=150, λ=0.2, m=5, p=0.03, p_{in}=0.3, p_{out}=0.075, w=0.8, initial composition: 0.2% cooperators, 0.4% free-riders, 0.4% actively selfish peers.

Summary. Simulation results prove that the audit-based approach is able to successfully detect and subsequently punish selfish peers by denying them the usage of the storage facility. Reputation may also have this capability if the replication rate is low. Even though, there are some false positives that may cause some cooperative peers being denied storage. Results reveal also the situations (such as loaded storage system and high replication rate) where the sliest selfish peers (who store data for some time and then destroy them) are circumventing the reputation or audit-based approach in order to be able to consume storage in the system without fulfilling their equal share.

4.3 Security considerations

In this section, we evaluate the robustness of reputation and audit-based mechanisms against the attacks exposed in 2.4.

Lying observers have no impact on the 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 any 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, and then peers are able to refuse storage to non cooperating peers, hence limiting free-riders.

5. CONCLUSION

We compared conventional reputation to an audit-based mechanism for P2P storage systems in which peer's observations originate from periodic verifications of data stored in the system. We demonstrated that the audit-based solution is more robust to selfish behavior than reputation. Therefore, we suggest that the former approach could be a good option for today's commercial storage systems. The reason behind this choice is the economic compensation peers acquire for storing data which encourages them to give false recommendations for fame. Additionally, we proposed a group-based design for audits management that may fit several types of networks such as social networks.

6. ACKNOWLEDGMENTS

The work presented here has been partially funded by Institut Telecom Initiative program on autonomic and spontaneous networks, the project SPREADS (ANR) and the ReSIST IST-026764 NoE.

7. REFERENCES

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

[2] AllMyData Tahoe: http://allmydata.org/trac/tahoe/

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

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

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

[6] 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.

[7] 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

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

[9] 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.

[10] Roger R. Dingledine, "The Free Haven project: Design and deployment of an anonymous secure data haven", Master's thesis, MIT, June 2000.

[11] 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.

[12] 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.

[13] 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.

[14] 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.

[15] 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

[16] 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/

[17] Emil Sit and Robert Morris, "Security Considerations for P2P Distributed Hash Tables", IPTPS 2002.

[18] 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.

[19] Michael Piatek, Tomas Isdal, Thomas Anderson, and Arvind Krishnamurthy, "Do incentives build robustness in BitTorrent?", In *Proceedings of the ACM/USENIX Fourth Symposium on Networked Systems Design and Implementation* (NSDI 2007), 2007.

[20] Robin I. M. Dunbar, "Co-Evolution of Neocortex Size, Group Size and Language in Human", *Behavioral and Brain Sciences* 16 (1993), no. 4, pp. 681-735.