Dependability Evaluation of Cluster-based Systems

Emmanuelle Anceaume, Francisco Brasileiro, Romaric Ludinard, Bruno Sericola, Frédéric Tronel

Publications Internes de l'IRISA
ISSN : 2102-6327
PI 1947 – Mars 2010

Dependability Evaluation of Cluster-based Systems

Emmanuelle Anceaume[*] , Francisco Brasileiro[**] , Romaric Ludinard[***] , Bruno Sericola[****] , Frédéric Tronel[*****]

Abstract: Awerbuch and Scheideler have shown that peer-to-peer overlays networks can survive Byzantine attacks only if malicious nodes are not able to predict what will be the topology of the network for a given sequence of join and leave operations. In this paper we investigate adversarial strategies by following specific protocols. Our analysis demonstrates first that an adversary can very quickly subvert distributed hash tables based overlays by simply never triggering leave operations. We then show that when all nodes (honest and malicious ones) are imposed on a limited lifetime, the system eventually reaches a stationary regime where the ratio of polluted clusters is bounded, independently from the initial amount of corruption in the system.

Key-words: Clusterized P2P Overlays, Adversary, Churn, Collusion, Markov chain.

[*] IRISA CNRS, anceaume@irisa.fr
[**] Universidade Federal de Campina Grande, LSD Laboratory, Campina Grande, Brazil, fubica@dsc.ufcg.edu.br
[***] INRIA Rennes Bretagne-Atlantique, romaric.ludinard@inria.fr, Supported by the Direction Générale des Entreprises - P2Pim@ges project
[****] INRIA Rennes Bretagne-Atlantique,bruno.sericola@inria.fr
[*****] Supélec, Rennes,ftronel@rennes.supelec.fr

1 Introduction

The adoption of peer-to-peer overlay networks as a building block for architecting Internet scale systems has raised the attention of making these overlays resilient not only to benign crashes, but also to malicious failure models [1, 2, 3]. As a result, Byzantine-resilient overlays have been proposed (e.g., [4, 5, 6, 7]). Awerbuch and Scheideler [8] have shown that peer-to-peer overlays networks can survive Byzantine attacks only if malicious nodes are not able to predict what will be the topology of the network for a given sequence of join and leave operations. A prerequisite for this condition to hold is to guarantee that nodes identifiers randomness is continuously preserved. However this is not sufficient as targeted join/leave attacks may quickly reveal the topology of the system [9]. Inducing churn has been shown to be the other fundamental ingredient to preserve randomness. Several strategies based on these principles have been proposed. Most of them are based on locally induced churn. However either they have been proven incorrect or they involve a too high level of complexity to be practically acceptable [8]. The other ones, based on globally induced churn, enforce limited lifetime for each node in the system. This keeps the system in an unnecessary hyper-activity, and thus need to impose strict restrictions on nodes joining rate which limit their applicability to open systems.

In this paper we propose to leverage the power of clustering to design a practically usable solution that preserves randomness under an adaptive adversary. Our solution, whose short version appears in [10], relies on the clusterized version of peer-to-peer overlays combined with a mechanism that allows the enforcement of limited nodes lifetime. Clusterized versions of structure-based overlays are such that nodes at the vertices of the graph are substituted by clusters of nodes. Cluster-based overlays have revealed to be well adapted for efficiently reducing the impact of churn on the system and/or in greatly reducing the damage caused by failures in the absence of targeted attacks [6, 11, 4].

The contributions of this paper are two-fold. We first investigate adversarial strategies by following specific protocols. Our analysis demonstrates that an adversary can very quickly subvert cluster-based overlays by simply never triggering leave operations. We then show that when all nodes are imposed on a limited lifetime, the system eventually reaches a stationary regime where the ratio of polluted clusters is bounded.

The remainder of this paper is organised as follows: In Section 2 we briefly describe the main features of cluster-based overlays, and the dependability issues. Section 3 presents the operations that provide access and departure from the system. In Section 4 we model adversarial behaviours through the use of protocols. We study the outcome of these protocols at cluster level by using a Markovian analysis. In this section, we consider a non restricted adversary. Section 5 is devoted to the same study in the case of a restricted adversary. Finally, Section 6 shows that by inducing churn at all peers of the system, safety of the system is preserved. We conclude in Section 7.

2 Cluster-based Overlays Networks

2.1 Structured Overlays Networks

An overlay network is a virtual network built on top of a physical network. Nodes of the overlay, usually called peers, communicate among each other along the edges of the overlay by using the communication primitives provided by the underlying network (e.g., IP network service). The algorithms that peers use to choose their neighbours and to route messages define the overlay topology. The topology of unstructured overlays conforms with random graphs, i.e., relationships among peers are mostly set according to a random process, and routing is not constrained. Structured overlays (also called Distributed Hash Tables (DHTs)) build their topology according to structured graphs (e.g., hypercube, torus). For most of them, the following principles hold: each peer is assigned a unique random identifier from an m-bit identifiers space. Identifiers (denoted IDs in the following) are derived by applying some standard cryptographic one-way hash function (e.g., MD5 hash function [12]) on the peers network address. The value of m (128 for standard MD5 function) is large enough to make the probability of identifiers collision negligible. The identifier space is partitioned among all the peers of the overlay. Peers self-organise within the graph according to a distance function D based on peers IDs (e.g., two peers are neighbours if their IDs share some common prefix), plus possibly other criteria such as geographical distance. Each application-specific object, or data-item, is assigned a unique identifier, called *key*, selected from the same m-bit identifier space. Each peer p owns a fraction of all the data items of the system. The mapping derives from the distance function D. In the following, we will use the term *peer* (or *key*) to refer to both the peer (or key) and its m-bit representation.

Following the seminal work of Plaxton et al [13], diverse DHTs have been proposed (e.g., CAN [14], Chord [15], Pastry [16], Kademlia [17]). All these DHTs have been proven to be highly satisfactory in terms of efficiency and scalability (i.e. their key-based routing interface guarantees operations whose complexity in messages and latency usually scale logarithmically with system size). However, in presence of adversarial behavior, relying on single peers to ensure the system connectivity and the correct retrieval of data is clearly not sufficient. Rather, by having peers gathered into quorums of peers (or equivalently clusters), peers may mutually supervise themselves, and run agreement protocols to lead to reliable operations. This has led to cluster-based structured overlays networks.

2.2 Cluster-based Structured Overlays Networks

Cluster-based structured overlays networks are such that clusters of peers substitute peers at the vertices of the graph. Each vertex of the graph is composed of a set of peers (also called *cluster* of peers). Peers join the clusters according to a given distance metric D. For instance in PeerCube [6], peer p joins the (unique) cluster whose label is a prefix of p's identifier, while in eQuus [11], p joins the (unique) cluster whose members are geographically the closest to p. Clusters in the system are uniquely labelled. Size of each cluster is both lower and upper bounded. The lower bound, named c in the following, usually satisfies some constraint based on the assumed failure model. For instance $c \geq 4$ allows Byzantine tolerant agreement protocols to be run among these c peers despite the presence of one Byzantine peer [18]. The upper bound, that we call s, is typically in $\mathcal{O}(\log N)$ where N is the current number of peers in the system, to meet scalability requirements. Once a cluster size exceeds s, this cluster `splits` into two smaller clusters, each one populated with the peers that are closer to each other according to distance D. Once a cluster undershoots its minimal size c, this cluster `merges` with the closest cluster in its neighbourhood. For space reasons we do not give any detail regarding the localisation of a cluster nor its `creation/split/merge` processes. None of these operations are necessary for the understanding of our work. The interested reader is invited to read their descriptions in the original papers (e.g. [6, 11, 4]).

In the present work we assume that at cluster level, peers are organised as core and spare members. Members of the core set are primarily responsible for handling messages routing and clusters operations. Management of the core set is such that its size is maintained to constant c. Spare members are the complement number of peers in the cluster. In contrast to core members, they are not involved in any of the overlay operations. Rationale of this classification is two-fold: first it allows to introduce the unpredictability required to deal with Byzantine attacks through a randomized core set generation algorithm (as shown in the sequel). Second it limits the management overhead caused by the natural churn present in typical overlay networks through the spare set management. In the following we assume that join and leave events have an equal chance to occur in any cluster.

2.3 Dependability Issues

A fundamental issue faced by any practical open system is the inevitable presence of peers that try to manipulate the system by exhibiting undesirable behaviours [3]. Such peers are classically called *malicious* or *Byzantine*. Malicious peers can devise complex strategies to prevent peers from discovering the correct mapping between peers and data keys. They can mount *Sybil attacks* [19] (i.e., an attacker generates numerous fake peers to pollute the system), they can do *routing-table poisoning* (also called *eclipse attacks* [1, 3]) by having good peers redirecting outgoing links towards malicious ones, or they can simply drop or re-route messages towards other malicious peers. They can magnify their impact by colluding and coordinating their behaviour. We model these strategies through a strong adversary that controls these malicious peers. We assume that the adversary has bounded resources in that it cannot control more than a fraction μ, $(0 < \mu < 1)$, of malicious peers in the whole network. Note that in the following we make a difference between the whole network and the overlay. The network encompasses all the peers that at some point may participate to the overlay (i.e. 2^m peers), while the overlay contains at any time a subset of these peers. Thus, while μ represents the assumed fraction of malicious peers in the network, the goal of the adversary is to populate the overlay with a larger fraction of malicious peers in order to subvert its functioning. Finally, a peer which always follows the prescribed protocols is said to be *honest*. Note that honest peers cannot tell *a priori* the difference between honest and malicious peers.

2.4 Security Schemes

We assume the existence of a public key cryptography scheme that allows each peer to verify the signature of each other peer. We also assume that correct peers never reveal their private keys. Peers IDs and keys are part of their coded state, and are acquired via a central authority [20]. When describing the protocols, we ignore the fact that messages are signed, and recipients of a message ignore any message that is not signed properly. We also use cryptographic techniques to prevent a malicious peer from observing or unnoticeably modifying a message sent by a correct peer. However a malicious peer has complete control over the messages it sends and receives. Note that messages physically sent between any two correct peers are neither lost nor duplicated.

3 Operations of the Overlay

When a peer joins a cluster or leaves it, corresponding operations are executed. Design of these operations takes advantage of peers role separation. The `leave` operation for peers in the core set introduces a certain amount of unpredictability required to deal with Byzantine attacks through a randomized core set generation algorithm. On the other hand both `leave` operations for peers in the spare set and `join` operations have no impact on the overall topology (provided that the size of the concerned cluster does not reach its lower or upper bounds). Specifically,

```
/* Protocol 1 */
/* stage 1 */
draw a ball b_0 from C ∪ S
/* stage 2 */
throw b_0 into the bag
if b_0 ∈ C then
   draw a ball b_1 from S
   and throw it into C
endif
draw a ball b_2 from the bag
and throw it into S
```

```
/* Protocol 2 */
/* stage 1 */
draw a ball b_0 from C ∪ S
/* stage 2 */
throw b_0 into the bag
if b_0 ∈ C then
  draw c balls from S ∪ C
  and throw these c balls into C
endif
draw a ball b_2 from the bag
and throw it into S
```

Figure 1: Rules of Protocol 1 and Protocol 2.

- `join`(p): When peer p `joins` a cluster, it joins it as a spare member.
- `leave`(p): When a peer p `leaves` a cluster either p belongs to the spare set or to the core set. In the former case, core members simply update their spare view to reflect p's departure, while in the latter case, the core view maintenance procedure is triggered. Two different maintenance protocols are analysed. The first one, referred to as *protocol 1* in the following, simply consists in replacing the left core member by one randomly chosen among spare members. The second one, referred to as *protocol 2* hereafter, consists in refreshing the whole core set by choosing c random peers within the whole cluster.

4 Model of the Adversarial Strategy

In this section, we investigate the two previously described protocols. Both protocols intend to prevent the adversary from elaborating deterministic strategies to win. These protocols are played in the following context. The very large number of peers in the network whose a fraction μ exhibits a malicious behavior is represented by a potentially infinite number of balls in a bag, with a proportion μ of red balls (malicious peers) and a proportion $1-\mu$ of white balls (honest peers), $\mu \in (0,1)$. Red and white balls are indistinguishable. Red balls are owned by the adversary. In addition to the bag, there are two urns, named $\mathcal{C}$ and $\mathcal{S}$ which respectively model the core set and the spare set of a cluster. Each protocol is a succession of rounds $r_1, r_2, \ldots$ during which the protocol rules described in Figure 1 are applied. Rules are oblivious to the colour of the balls, that is, they cannot distinguish between the white and the red balls.

The goal of the adversary is to get a quorum of red balls in both urns $\mathcal{C}$ and $\mathcal{S}$ so that the number of red balls in $\mathcal{C}$ is bound to always exceed $\lfloor (c-1)/3 \rfloor$ [18]. The adversary may at any time inspect both urns and bag to elaborate adversarial strategies to win over the protocol. In particular it may not follow the rules prescribed by the protocols by preventing its red balls from being extracted from both urns. Specifically, at stage 1 of both protocols, if the drawn ball b_0 is red then the adversary puts the ball back into the urn it was drawn from. In that case, stage 2 is not applied, and a new round is triggered. Clearly this strategy ensures that the number of red balls in $\mathcal{C} \cup \mathcal{S}$ is non decreasing.

We model the effects of these rounds using a homogeneous Markov chain denoted by $X = \{X_n, n \geq 0\}$ representing the evolution of the number of red balls in both urns $\mathcal{C}$ and $\mathcal{S}$. More formally, the state space Ω of X is defined by $\Omega = \{(x,y) \mid 0 \leq x \leq c,\ 0 \leq y \leq s\}$, and, for $n \geq 1$, the event $X_n = (x,y)$ means that, after the n-th transition or n-th round, the number of red balls into urn $\mathcal{C}$ is equal to x and the number of red balls into urn $\mathcal{S}$ is equal to y. The transition probability matrix P of X depends on both the rules of the given protocol and the adversarial behaviour. This matrix is detailed in each of the following subsections. We define a state as *polluted* if in this state urn $\mathcal{C}$ contains strictly more than $\lfloor (c-1)/3 \rfloor$ balls. In the following, we denote by c' the value $\lfloor (c-1)/3 \rfloor$. Conversely, a state that is not polluted is said to be *safe*. In the remaining of this paper, the initial probability distribution is denoted by α.

4.1 First Protocol

Regarding the first protocol, the subset of safe states A is defined as: $A = \{(x,y) \mid 0 \leq x \leq c',\ 0 \leq y \leq s\}$, while the set of polluted states B, is the subset $\Omega - A$, i.e. $B = \{(x,y) \mid c' < x \leq c,\ 0 \leq y \leq s\}$. By the rules of the protocol, one can never escape from the subset of states B to switch back to safe states A since the number of red balls in $\mathcal{C}$ is non decreasing. Thus, the adversary wins the protocol when the process X reaches the closed subset B.

Figure 2 illustrates the states partition of the process X. Matrix P and the initial probability vector α are partitioned in a manner that matches the decomposition of $\Omega = A \cup B$, that is

$$P = \begin{pmatrix} P_A & P_{AB} \\ 0 & P_B \end{pmatrix} \text{ and } \alpha = (\alpha_A \ \ \alpha_B).$$

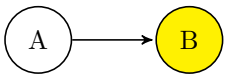


Figure 2: Aggregated view of the Markov chains associated with protocol 1. Safe states are represented by A, and polluted ones by B. B is an absorbing class.

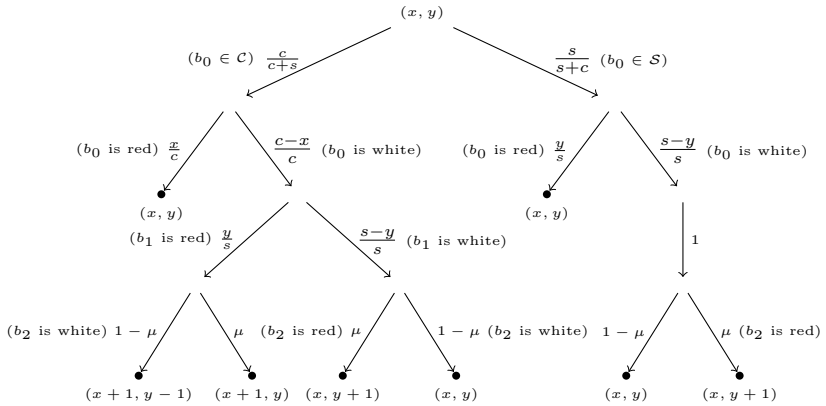


Figure 3: Transition diagram for the computation of the transition probability matrix P of Protocol 1.

where P_A (resp. P_B) is the sub-matrix of dimension $|A| \times |A|$ (resp. $|B| \times |B|$), containing the transitions between states of A (resp. B); P_{AB} is the sub-matrix of dimension $|A| \times |B|$, containing the transitions from states of A to states of B; B is an absorbing class thus $P_{BA} = 0$. Finally, sub-vector α_A (resp. α_B) contains the initial probabilities of states of A (resp. B).

The computation of the transition probabilities of the matrix P is illustrated in Figure 3. In this tree, each edge is labelled by a probability and its corresponding event according to the rules of the protocol (see Figure 1). This figure can be interpreted as follows: At round r, $r \geq 1$, starting from state (x, y) (root of the tree), the Markov chain can visit four different states, namely (x, y), $(x, y+1)$, $(x+1, y)$, and $(x+1, y+1)$ (leaves of the tree). The probability associated with each one of these transitions is obtained by summing the products of the probabilities discovered along each path starting from the root to the leaf corresponding to the target state. We derive the transition probability matrix P of the Markov chain X associated with this protocol. For all $x \in \{0, \ldots, c\}$ and for all $y \in \{0, \ldots, s\}$, we have

$$
\begin{aligned}
p_{(x,y),(x,y)} &= \left(\frac{1}{c+s}\right)\left(x + \frac{(c-x)(s-y)(1-\mu)}{s} + y + (s-y)(1-\mu)\right) \\
p_{(x,y),(x,y+1)} &= \frac{(c+s-x)(s-y)\mu}{(c+s)s} \quad \text{for } y \leq s-1 \\
p_{(x,y),(x+1,y-1)} &= \frac{(c-x)y(1-\mu)}{(c+s)s} \quad \text{for } x \leq c-1 \text{ and } y \geq 1 \\
p_{(x,y),(x+1,y)} &= \frac{(c-x)y\mu}{(c+s)s} \quad \text{for } x \leq c-1.
\end{aligned}
$$

In all other cases, transition probabilities are null.

4.2 Second Protocol

For $s = 1$, it is easy to see that the second protocol is equivalent to the first one. On the other hand, in contrast to the first protocol, the second protocol alternates between safe and polluted states. After a random number of these alternations the process ends by entering a set of closed polluted states. Indeed, by the rules of the protocol, one can escape finitely often from polluted states (x, y) to switch back to safe states as long as (x, y) satisfies $c' < x + y \leq s + c'$ (there are still sufficiently many white balls in both $\mathcal{C}$ and $\mathcal{S}$ so as to successfully withdrawing c balls such that $\mathcal{C}$ is safe). However, there is a round such that state (x, y) is entered, where $x + y > s + c'$. From this round onwards, going back to safe states is impossible: the adversary has won the protocol. Formally, the subset of safe states A is defined, as for the first protocol, by $A = \{(x, y) \mid 0 \leq x \leq c',\ 0 \leq y \leq s\}$. On the other hand we need to decompose the set B of polluted states into two subsets C and D defined by $C = \{(x, y) \mid c' < x + y \leq s + c',\ c' < x < c,\ 0 \leq y \leq s,\ s > 1\}$, and $D = \{(x, y) \mid (x + y > s + c',\ s > 1) \vee (x > c',\ s = 1),\ 0 \leq y \leq s\}$. Subsets A and C are transient and subset D is a closed subset. Figure 4 illustrates the states partition of the process X.

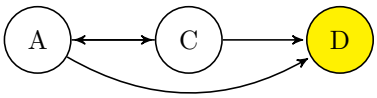


Figure 4: Aggregated view of the Markov chains associated with protocol 2. Safe states are represented by A, and polluted states by C and D. State D is an absorbing class.

Following the decomposition of $\Omega = A \cup C \cup D$, we partition matrix P and the initial probability vector α by writing

$$P = \begin{pmatrix} P_A & P_{AC} & P_{AD} \\ P_{CA} & P_C & P_{CD} \\ 0 & 0 & P_D \end{pmatrix} \text{ and } \alpha = (\alpha_A \;\; \alpha_C \;\; \alpha_D).$$

By proceeding similarly as above, we can derive the transitions of process X associated with protocol 2. Briefly, when the protocol starts in state (x, y) at round r, it remains in state (x, y) during the round if either ball b_0 is red or b_0 is white, it has been drawn from $\mathcal{S}$, and b_2 is white. It changes to state $(x, y + 1)$ if b_0 is white, it has been drawn from $\mathcal{S}$, and b_2 is red. Finally the protocol switches to state $(k, x + y - k + \ell)$, where k is an integer $k = 0, \ldots, \min(c, x + y)$ and $\ell = 0$ or 1 if b_0 is white, it has been drawn from $\mathcal{C}$, and the renewal process leads to the choice of k red balls. For all $x \in \{0, \ldots, c\}$, $y \in \{0, \ldots, s\}$, and $s > 1$, we have

$$\begin{aligned}
p_{(x,y),(x,y)} &= \frac{x}{c+s} + \frac{y}{c+s} + \frac{s-y}{c+s}(1-\mu) + \frac{c-x}{c+s}(1-\mu)q(x, x+y) \\
p_{(x,y),(x,y+1)} &= \frac{s-y}{c+s}\mu + \frac{c-x}{c+s}\mu q(x, x+y) \text{ for } y \leq s-1 \\
p_{(x,y),(k,x+y-k)} &= \left(\frac{c-x}{c+s}\right)(1-\mu)\, q(k, x+y) \text{ for } 0 \leq k \leq \min(c, x+y) \text{ and } k \neq x \\
p_{(x,y),(k,x+y-k+1)} &= \left(\frac{c-x}{c+s}\right)\mu q(k, x+y) \text{ for } 0 \leq k \leq \min(c, x+y) \text{ and } k \neq x
\end{aligned}$$

where

$$q(z, x+y) = \frac{\binom{x+y}{z}\binom{c+s-1-(x+y)}{c-z}}{\binom{c+s-1}{c}} 1_{\{0 \leq x+y-z \leq s-1\}} \tag{1}$$

is the probability of getting z red balls when c balls are drawn, without replacement, in an urn containing $x + y$ red balls and $c + s - 1 - (x + y)$ white balls, referred to as the hypergeometric distribution. $1_{\{\ldots\}}$ represents the indicator function. In all other cases, transition probabilities are null.

4.3 Study of both Protocols in an Adversarial Environment

As described in Section 4.1 the Markov chain X associated with protocol 1 is reducible and the states of A are transient, which means that matrix $I - P_A$ is invertible, where I is the identity matrix of dimension $|A|$. Recall that B is an absorbing class, i.e. $P_{BA} = 0$. Similarly, as described in Section 4.2 the Markov chain X associated with protocol 2 is reducible, the states of A and C are transient and subset D is a closed subset. We start our study by first investigating for $i = 1, 2$ the distribution $T_A^{(i)}$ which counts the total number of rounds spent by protocol i in the subset of safe states A before absorption. Specifically $T_A^{(1)} = \inf\{n \geq 0 \mid X_n \in B\}$. The probability mass function of T_A^1 for $k \geq 0$ is easily derived as

$$\mathbb{P}\{T_A^{(1)} = k\} = \begin{cases} \alpha_B \mathbb{1} & \text{if } k = 0 \\ \alpha_A (P_A)^{k-1}(I - P_A)\mathbb{1} & \text{if } k \geq 1 \end{cases}$$

where $\mathbb{1}$ is the column vector with all components equal to 1. The cumulative distribution function and the expectation of $T_A^{(1)}$ are respectively given by

$$\mathbb{P}\{T_A^{(1)} \leq k\} = 1 - \alpha_A (P_A)^k \mathbb{1}, \text{ and } E(T_A^{(1)}) = \alpha_A (I - P_A)^{-1}\mathbb{1}. \tag{2}$$

Following the results obtained in Sericola [21], for the second protocol, the probability mass function of $T_A^{(2)} = \sum_{n=1}^{\infty} 1_{\{X_n \in A\}}$ is given by

$$\mathbb{P}\{T_A^{(2)} = k\} = \begin{cases} 1 - v\mathbb{1} & \text{if } k = 0 \\ vR^{k-1}(I - R)\mathbb{1} & \text{if } k \geq 1 \end{cases} \tag{3}$$

where $v = \alpha_A + \alpha_C(I - P_C)^{-1}P_{CA}$ and $R = P_A + P_{AC}(I - P_C)^{-1}P_{CA}$.

Its cumulative distribution function and its expectation are respectively given by

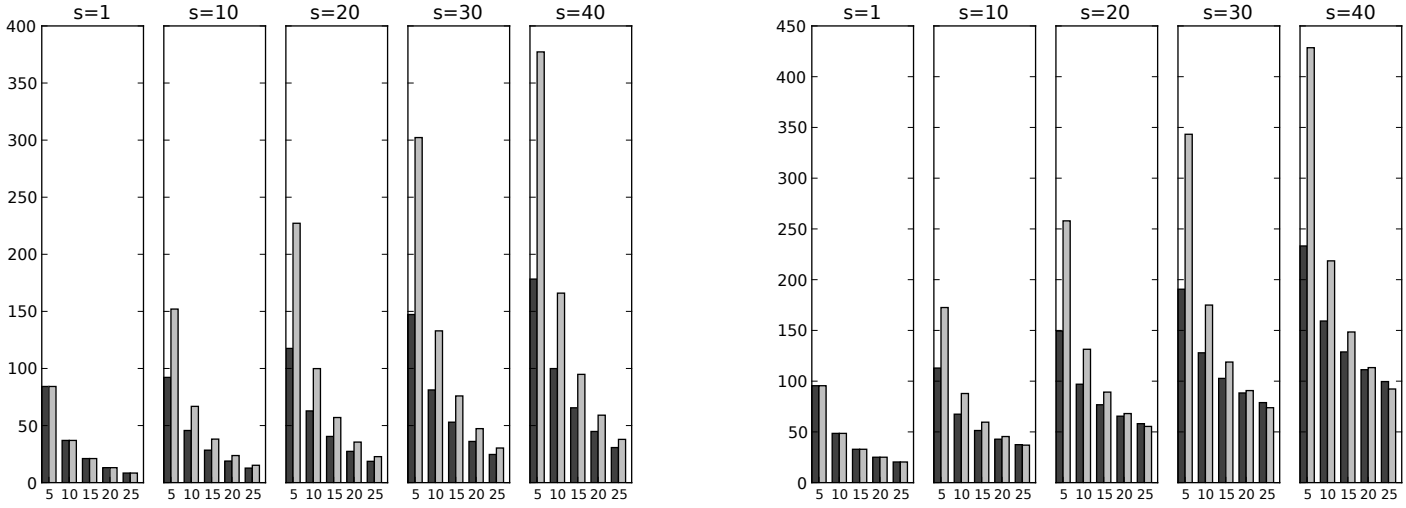
$$\mathbb{P}\{T_A^{(2)} \leq k\} = 1 - vR^k\mathbb{1}, \text{ and } E(T_A^{(2)}) = v(I - R)^{-1}\mathbb{1}. \tag{4}$$

In the experiments run for this work, we consider two initial distributions. The first one, which we denote by β, consists in drawing $c + s$ balls from the bag such that c of them are thrown into urn $\mathcal{C}$, and the other s ones are thrown into urn $\mathcal{S}$. This initial state X_0 is defined by $X_0 = (C_r, S_r)$ where C_r (resp. S_r) is the initial number of red balls in $\mathcal{C}$ (resp. $\mathcal{S}$). Thus both C_r and S_r follow a binomial distribution, and assuming they are independent, we get, for $x = 0, \ldots, c$ and $y = 0, \ldots, s$,

$$\begin{aligned} \beta(x, y) &= \mathbb{P}\{C_r = x, S_r = y\} = \mathbb{P}\{C_r = x\}\mathbb{P}\{S_r = y\} \\ &= \binom{c}{x}\mu^x(1-\mu)^{c-x}\binom{s}{y}\mu^y(1-\mu)^{s-y}. \end{aligned} \tag{5}$$

The second one, that we denote by δ, consists simply in starting from state $(0, 0)$, that is the state free from red balls, i.e.

$$\delta(x, y) = \delta_{0x}\delta_{0y} \text{ with } \delta_{ij} \text{ the Kronecker delta.} \tag{6}$$



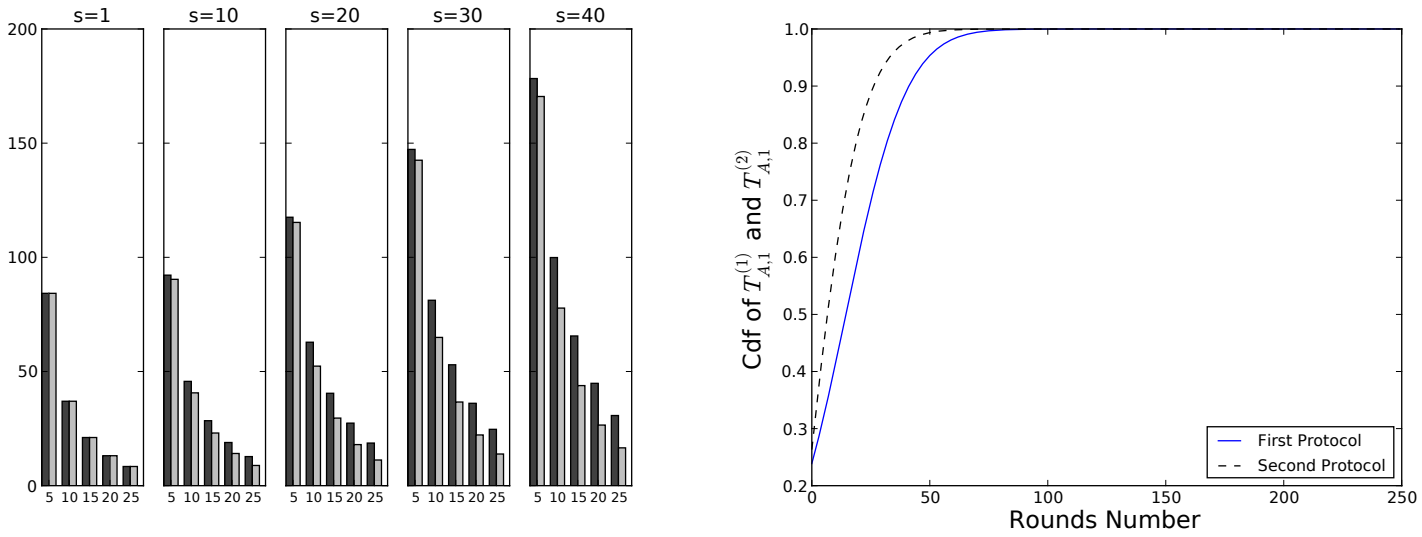
(a) $\alpha = \beta$ (see Relation (5))

(b) $\alpha = \delta$ (see Relation (6))

Figure 5: $E(T_A^{(1)})$ (see Relation (2)) represented by dark bars and $E(T_A^{(2)})$ (see Relation (4)) represented by light grey bars as a function of s and the percentage μ of red balls. Notation $5, 10, \ldots 25$ denotes $\mu = 5\%, 10\%, \ldots 25\%$. In all these experiments, $c = 10$.

Figure 5(a) (resp. Figure 5(b)) compares the expected number of rounds run in safe states for both protocols. In accordance with the intuition, increasing the size of the urns augments both $E(T_A^{(1)})$ and $E(T_A^{(2)})$ independently from the ratio of red balls in the bag. Similarly, for a given cluster size, augmenting the ratio of red balls in the bag decreases both $E(T_A^{(1)})$ and $E(T_A^{(2)})$. Now Figure 5(b) shows that for small values of μ the second protocol overpasses the first one, while for larger values of μ, protocol 1 is better than protocol 2. Interpretation of this result is as follows. When the size s of $\mathcal{S}$ is equal to 1, both protocols are equivalent as illustrated in Figure 5. Now, consider the case where the size s of $\mathcal{S}$ gets larger compared to $\mathcal{C}$'s one. The probability to draw a ball from $\mathcal{S}$ tends to 1. Now as the adversary never withdraws its red balls from any urns, the number of red balls within $\mathcal{S}$ is non decreasing. Hence, the larger μ is the faster the ratio of red balls in $\mathcal{S}$ tends to 1. With small probability, a ball from $\mathcal{C}$ is drawn. In protocol 1 it is replaced with high probability by a red ball drawn from $\mathcal{S}$. Hence to reach a polluted state, $c' + 1$ white balls have to be replaced by red ones. While with protocol 2 the renewal of $\mathcal{C}$ reaches with high probability a polluted state in a single step. Thus, even if the second protocol continues to alternate for a finite number of times between safe and polluted subset of states, the total time spent in safe states is less than the one spent by the first protocol. Note that one cannot observe such a behavior when the initial ratio of red balls in both urns is non null (as shown in Figure 5(a)) as it takes less steps for both protocols to switch to polluted states.

A deeper investigation of the second protocol behavior can be done by studying the duration and frequency of successive sojourn times in subsets A and C. For $n \geq 1$, we denote by $T_{A,n}^{(2)}$ (respectively $T_{C,n}^{(2)}$) the distribution of the time spent by



(a) Initial probability vector $\alpha = \beta$, and $c = 10$.

(b) $\alpha = \beta$, $c = 10$, $s = 20$, and μ=0.25.

Figure 6: First sojourn time in transient states. a) $E(T_{A,1}^{(1)})$ represented by dark bars (Relation (2)) and $E(T_{A,1}^{(2)})$ represented by light grey bars (Relation (9)) as a function of s and the percentage of red balls in the bag. b) $\mathbb{P}\{T_{A,1}^{(1)} \leq k\}$ (Relation (2)) and $\mathbb{P}\{T_{A,1}^{(2)} \leq k\}$ (Relation (7)) as a function of the rounds number.

the Markov chain X during its n-th sojourn in subset A (resp. C). Thus the total time spent in subset A (resp. C) before reaching subset D is given by

$$T_A^{(2)} = \sum_{n=1}^{\infty} T_{A,n}^{(2)} \text{ and } T_C^{(2)} = \sum_{n=1}^{\infty} T_{C,n}^{(2)}$$

From Sericola and Rubino. [22], we have for $n \geq 1$ and $k \geq 0$

$$\mathbb{P}\{T_{A,n}^{(2)} \leq k\} = 1 - vG^{n-1}(P_A)^k \mathbb{1}, \tag{7}$$

where v is defined in Relation (3) and $G = (I - P_A)^{-1} P_{AC} (I - P_C)^{-1} P_{CA}$.
Symmetrically, we have

$$\mathbb{P}\{T_{C,n}^{(2)} \leq k\} = 1 - wH^{n-1}(P_C)^k \mathbb{1}, \tag{8}$$

where $w = \alpha_C + \alpha_A (I - P_A)^{-1} P_{AC}$ and $H = (I - P_C)^{-1} P_{CA} (I - P_A)^{-1} P_{AC}$.

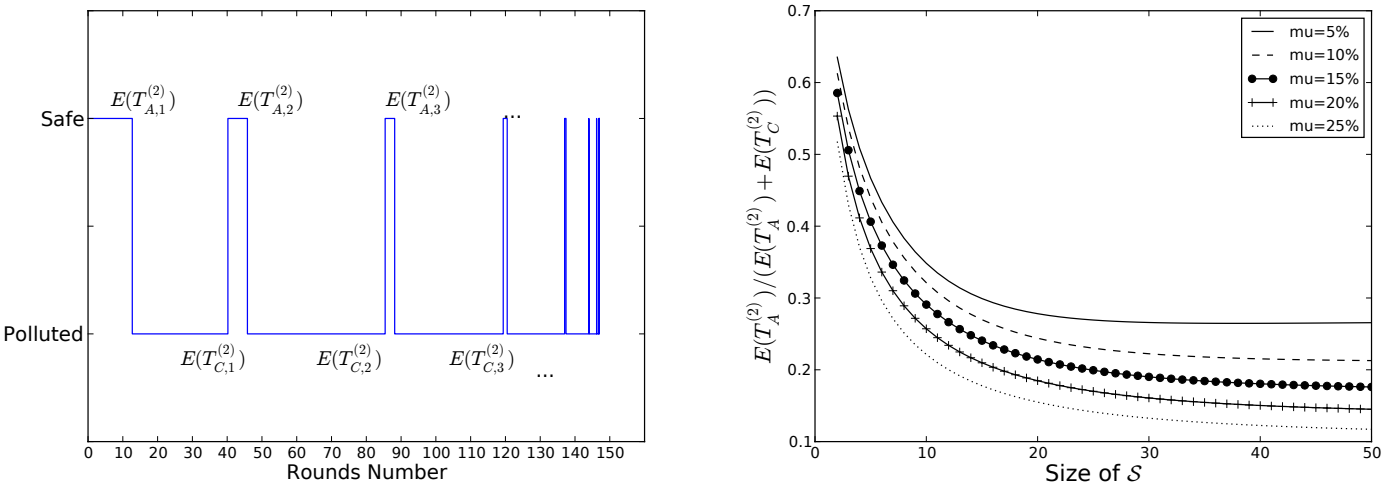
The expectations of $T_{A,n}^{(2)}$ and $T_{C,n}^{(2)}$ are respectively given by

$$E(T_{A,n}^{(2)}) = vG^{n-1}(I - P_A)^{-1}\mathbb{1} \text{ and } E(T_{C,n}^{(2)}) = wH^{n-1}(I - P_C)^{-1}\mathbb{1}. \tag{9}$$

Before investigating the alternation between safe and polluted states, we first observe the behavior of both protocols during their first sojourn time in the subset of safe states A. Recall that by construction of protocol 1, $T_{A,1}^{(1)} = T_A^{(1)}$. Similarly to what has been observed previously, Figure 6(a) shows that increasing the size of the urns augments the expected duration of the first sojourn time in A independently from the ratio μ of red balls in the bag. Similarly, for a given cluster size s, increasing μ decreases $E(T_{A,1}^{(1)})$ and $E(T_{A,1}^{(2)})$. On the other hand, in contrast to what has been observed in Figure 5 the first protocol always overpasses the second one regarding the duration of the first sojourn time in safe states, both in expectation and with respect to their cumulative distribution functions (see Figure 6(b)). This result emphasises what has been observed in Figure 5(b) for $\mu = 25\%$.

Figure 7(a) shows the expected duration and frequency of successive sojourn times in subsets A and C for protocol 2. Clearly, the protocol runs more rounds in C than it does in A. Note that for $n \geq 7$, the expected duration of the sojourn times in both A and C is already close to 0. Figure 7(b) depicts the percentage of rounds spent by protocol 2 in safe states before absorption as a function of the size s of $\mathcal{S}$. This percentage is described by

$$\frac{E(T_A^{(2)})}{E(T_A^{(2)}) + E(T_C^{(2)})} = \frac{v(I - R)^{-1}\mathbb{1}}{v(I - R)^{-1}\mathbb{1} + w(I - T)^{-1}\mathbb{1}} \tag{10}$$



(a) $\alpha = \beta$, $c = 10$, $s = 20$ and $\mu = 25\%$

(b) $\alpha = \beta$, and $c = 10$

Figure 7: Sojourn times in transient states. a) Succession of $E(T^{(2)}_{A,n})$ and $E(T^{(2)}_{C,n})$ as a function of the rounds number. b) $E(T_A^{(2)})/(E(T_A^{(2)}) + E(T_C^{(2)}))$ as a function s.

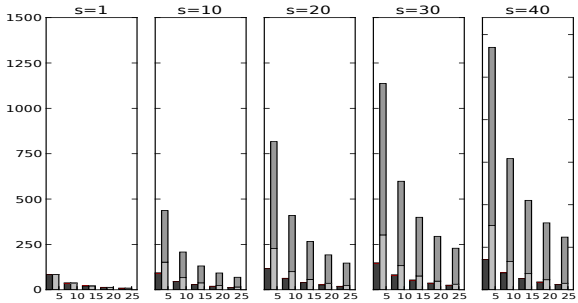


Figure 8: Expected total duration of transient states. $E(T_A^{(1)})$ (Relation (2)) represented by dark coloured bars, $E(T_A^{(2)}) + E(T_C^{(2)})$ (Relation (10)) represented by light grey and grey coloured bars as a function of s and the percentage of red balls in the bag. In all these experiments, $c = 10$.

where v and R are given in Relation (3), w in Relation (8), and $T = P_C + P_{CA}(I - P_A)^{-1}P_{AC}$.

In accordance with the intuition, this percentage decreases with larger values of μ, and stabilises for increasing values of s (which is confirmed by the behavior presented in Figure 7(a)). Finally an aggregated view of the total time spent in transient states, i.e. $A \cup C$, before absorption is depicted in Figure 8. The main observation drawn from this figure is that the total time spent in transient states linearly increases with the size of $\mathcal{S}$. On the other hand, this time is mainly spent in C, and this tendency increases with larger values of s, which confirms Figure 7(a).

The main lessons learnt from this preliminary study is that increasing the amount of randomisation of the protocol does make it necessarily more resilient to stronger adversaries. From a practical point of view this result is interesting as handling the renewal of a set of entities requires costly agreement protocols to be run among the interested parties. Typically, complexity of these protocols is in $\mathcal{O}(n^3)$ where n is the number of parties. Unfortunately by allowing malicious peers to stay infinitely long at the same position in the overlay, both protocols are not sufficient to prevent the adversary from progressively surrounding honest peers and gaining the quorum. In the following section we show that by limiting the lifetime of malicious peers in the same cluster, safety of the whole cluster is eventually guaranteed.

5 Constraining the Adversary by Limiting its Sojourn Time in a Cluster

It has been shown [9] that structured overlays cannot survive targeted attacks if the adversary may keep sufficiently long its malicious peers at the same position in the overlay. Indeed, once malicious peers have succeeded in sitting in a focused region

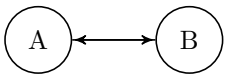


Figure 9: Aggregated view of the Markov chains associated with both protocols. Safe states are represented by A, and polluted ones by B.

of the overlay, they can progressively gain the quorum by simply waiting for honest peers to leave their position, leading to the eventual isolation of honest peers. The two fundamental properties that prevent peers isolation are firstly, the guarantee that peers identifiers are random, and secondly that peers cannot stay forever at the same position in the system [8].

From a practical point of view, implementing limited peers lifetime and unpredictable identifier assignment, can be achieved by adding an incarnation number to the fields that appear in the peer's certificate (certificate are acquired at trustworthy Certification Authorities (CAs)), and by hashing this certificate to generate the peer's identifier. By the properties of hash functions, we guarantee that peers identifiers are unpredictable. The incarnation number limits the lifetime of identifiers. The current incarnation k of any peer can be computed as $k = \lceil (t - ivt)/I \rceil$, where ivt is the initial validity time of the peer's certificate, t is the current time, and I is the length of the lifetime of each peer's incarnation. Thus, the k^{th} incarnation of a peer p expires when its local clock reads $ivt + kI$. At this time p must rejoin the system using its $(k+1)^{th}$ incarnation. The ivt is one of the fields in the peer's certificate and since certificates are signed by the CA, it cannot be unnoticeably modified by a malicious peer. Moreover, a certificate commonly contains the public key of the certified entity. This way, messages exchanged by the peers can be signed using the corresponding private key, preventing malicious peers from unnoticeably altering messages originated from other peers in the system. Messages must contain the certificate of their issuer, so as to allow recipients to validate them. Therefore, at any time, any peer can check the validity of the ID of any other peers in the system, by simply calculating the current incarnation of the other peer and generating the corresponding ID. If some peer detects that the ID of one of its neighbours is not valid then it cuts its connection with it.

Another solution is to periodically choose at random a peer in the overlay, invalidating its current identifier, so that a new one must be acquired via a CA to rejoin the overlay. Note that this second solution requires Byzantine agreement to be run among a possibly large subset of peers to elect a random peer, which may limit its practicability.

Coming back to our analysis, we model the constraint on the adversary by preventing the adversary from keeping its red balls in both urns, so that randomness among red and white balls is continuously preserved. As previously, we investigate both protocols presented in Figure 1. It is not difficult to see that both protocols alternate between the subset of safe states $A = \{(x, y) \mid 0 \leq x \leq c',\ 0 \leq y \leq s\}$, and the subset of polluted ones $B = \{(x, y) \mid c' < x \leq c,\ 0 \leq y \leq s\}$ for an infinite number of rounds. Both subsets A and B are transient. Figure 9 illustrates the states partition of the process X. Following the decomposition of $\Omega = A \cup B$, we partition matrix P and the initial probability vector α by writing

$$P = \begin{pmatrix} P_A & P_{AB} \\ P_{BA} & P_B \end{pmatrix} \text{ and } \alpha = (\alpha_A \ \ \alpha_B).$$

5.1 First Protocol

By proceeding exactly as in Section 4, we derive the transition probability matrix P for the first protocol. For all $x \in \{0, \ldots, c\}$ and $y \in \{0, \ldots, s\}$, the entries of P for the first protocol are given by

$$\begin{aligned}
p_{(x,y),(x,y)} &= \frac{xy + (c(s-y) - xs)(1-\mu)}{(c+s)s} + \frac{y\mu + (s-y)(1-\mu)}{c+s} \\
p_{(x,y),(x,y-1)} &= \frac{(x+s)y}{(c+s)s}(1-\mu) \text{ for } y \geq 1 \\
p_{(x,y),(x,y+1)} &= \left(\frac{c-x+s}{c+s}\right)\left(\frac{s-y}{s}\right)\mu \text{ for } y \leq s-1 \\
p_{(x,y),(x+1,y-1)} &= \frac{(c-x)y}{(c+s)s}(1-\mu) \text{ for } x \leq c-1 \text{ and } y \geq 1 \\
p_{(x,y),(x+1,y)} &= \frac{(c-x)y}{(c+s)s}\mu \text{ for } x \leq c-1 \\
p_{(x,y),(x-1,y)} &= \frac{x(s-y)}{(c+s)s}(1-\mu) \text{ for } x \geq 1 \\
p_{(x,y),(x-1,y+1)} &= \frac{x(s-y)}{(c+s)s}\mu \text{ for } x \geq 1 \text{ and } y \leq s-1.
\end{aligned} \tag{11}$$

In all other cases, transition probabilities are null.

5.2 Second Protocol

Similarly the entries of P for the second protocol are for all $x \in \{0,\ldots,c\}$ and $y \in \{0,\ldots,s\}$

$$\begin{aligned}
p_{(x,y),(x,y)} &= \frac{xq(x,x+y-1)\mu}{c+s} + \frac{(c-x)q(x,x+y)(1-\mu)}{c+s} + \frac{y\mu+(s-y)(1-\mu)}{c+s} \\
p_{(x,y),(x,y-1)} &= \frac{x}{c+s}q(x,x+y-1)(1-\mu) + \frac{y}{c+s}(1-\mu) \text{ for } y \geq 1 \\
p_{(x,y),(x,y+1)} &= \frac{c-x}{c+s}q(x,x+y)\mu + \frac{s-y}{c+s}\mu \text{ for } y \leq s-1 \\
p_{(x,y),(k,x+y-k-1)} &= \frac{x}{c+s}q(k,x+y-1)(1-\mu) \\
&\quad \text{for } \max(0,x+y-1-s) \leq k \leq \min(c,x+y-1) \text{ and } k \neq x \\
p_{(x,y),(k,x+y-k)} &= \frac{x}{c+s}q(k,x+y-1)\mu + \frac{c-x}{c+s}q(k,x+y)(1-\mu) \\
&\quad \text{for } \max(0,x+y-s) \leq k \leq \min(c,x+y-1) \text{ and } k \neq x \\
p_{(x,y),(k,x+y-k+1)} &= \frac{c-x}{c+s}q(k,x+y)\mu \\
&\quad \text{for } \max(0,x+y+1-s) \leq k \leq \min(c,x+y) \text{ and } k \neq x,
\end{aligned} \tag{12}$$

where $q(z,x+y)$ is given by Relation (1). In all other cases, transition probabilities are null.

5.3 Study of both Protocols in a Constrained Adversarial Environment

By proceeding as in Section 4.3, we investigate the behavior of both processes X during their successive n-th sojourn time in A and B. Each process X is irreducible and aperiodic since at least one state has a transition to itself. For $n \geq 1$, we denote by $T_{A,n}^{(i)}$ (resp. $T_{B,n}^{(i)}$) the time spent by the Markov chain X associated with protocol i during its n-th sojourn in subset A (resp. B). The expectations of $T_{A,n}^{(i)}$ and $T_{B,n}^{(i)}$ are respectively given for $i = 1, 2$ by

$$E(T_{A,n}^{(i)}) = vG^{n-1}(I-P_A)^{-1}\mathbb{1} \text{ and } E(T_{B,n}^{(i)}) = wH^{n-1}(I-P_B)^{-1}\mathbb{1} \tag{13}$$

where v is defined as in Relation (3), G as in Relation (7), and w and H are defined as in Relation (8).

For both protocols, the Markov chain X is finite, irreducible and aperiodic so the stationary distribution exists and is unique. We denote by π the stationary distribution of the Markov chain X. The row vector π is thus the unique solution to the linear system $\pi P = \pi$ and $\pi\mathbb{1} = 1$. As we did for row vector α, we partition vector π according to the partition $\Omega = A \cup B$, by writing $\pi = (\pi_A \quad \pi_B)$, where sub-vector π_A (resp. π_B) contains the stationary probabilities of states of A (resp. B). The mean percentage of time spent in subset A during the j-th sojourn is equal for protocol $i = 1, 2$ to $E(T_{A,j}^{(i)})/(E(T_{A,j}^{(i)}) + E(T_{B,j}^{(i)}))$. By Cesàro lemma,

$$\lim_{n\to\infty} \frac{\sum_{j=1}^{n} E(T_{A,j}^{(i)})}{\sum_{j=1}^{n} E(T_{A,j}^{(i)}) + E(T_{B,j}^{(i)})} = \lim_{n\to\infty} \frac{E(T_{A,n}^{(i)})}{E(T_{A,n}^{(i)}) + E(T_{B,n}^{(i)})} = \pi_A\mathbb{1}$$

Figure 10 shows the behaviour of the Markov chains associated with both protocols during their first 10 sojourn times in both A and B. The first observation is related to the percentage of time spent by Markov chains in safe states. As depicted in Figure 10.(a) both Markov chains tend to spend more than 2/3 of their time in safe states. Both convergence speeds are very fast (convergence is reached in a single alternation for the first Markov chain, and in 5 ones for the second one). The second remark concerns the frequency at which safe and polluted states alternate. Figure 10.(b) shows that this frequency is almost 3 times lower for the first process than for the second one. From a practical point of view this is interesting as it makes the first protocol more adapted to dynamic environment compared to the second one (i.e., protocol 1 handles a higher number of connections and disconnections before switching to a polluted state than protocol 2).

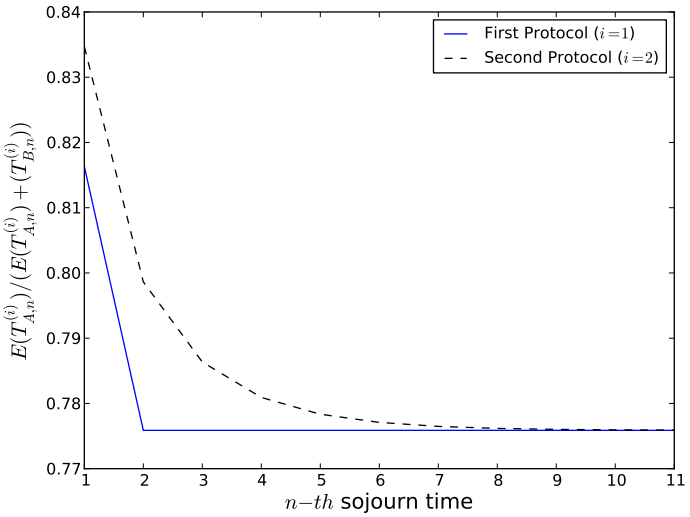
Theorem. 1 *For both protocols 1 and 2, the stationary distribution π is equal to β, i.e. for all $x = 0,\ldots,c$ and $y = 0,\ldots,s$, we have*

$$\lim_{n\longrightarrow\infty} \mathbb{P}\{X_n = (x,y)\} = \beta(x,y) \text{ with } \beta(x,y) \text{ given by relation (5).}$$

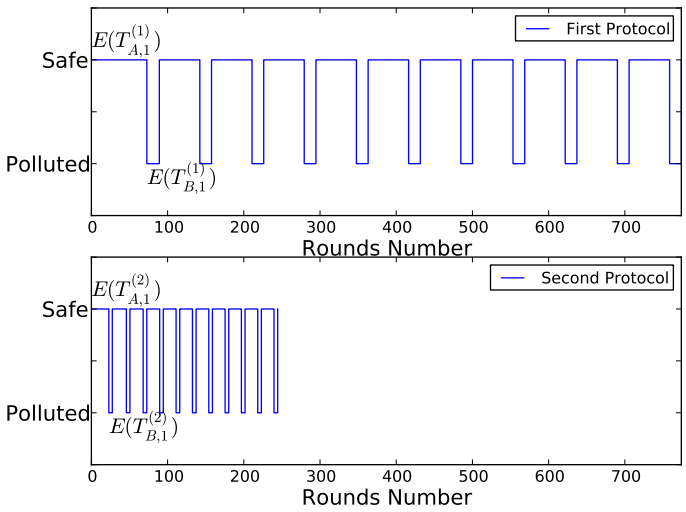
Proof

We show that for both protocols we have $\beta P = \beta$, i.e. for all $i \in \{0,\ldots,c\}$ and $j \in \{0,\ldots,s\}$, we have

$$(\beta P)(i,j) = \sum_{u=0}^{c}\sum_{v=0}^{s} \beta(u,v)p_{(u,v),(i,j)} = \beta(i,j).$$



(a) $E(T_{A,n}^{(i)})/(E(T_{A,n}^{(i)}) + E(T_{B,n}^{(i)}))$ as defined in Relation (13) as a function of the n-th sojourn time.



(b) Succession of $E(T_{A,n}^{(i)})$ and $E(T_{B,n}^{(i)})$ as a function of the rounds number.

Figure 10: Comparison of sojourn times in safe and polluted states. The initial probability vector α is equal to β (Relation 5), $c = 10$, $s = 20$ and $\mu = 25\%$.

First of all, note that, from relation (5), we have

$$\begin{array}{lcll}
\beta(i,j+1) & = & \beta(i,j)\frac{(s-j)\mu}{(j+1)(1-\mu)} & \text{for } j \leq s-1, \\
\beta(i,j-1) & = & \beta(i,j)\frac{j(1-\mu)}{(s-j+1)\mu} & \text{for } j \geq 1, \\
\beta(i-1,j+1) & = & \beta(i,j)\frac{i(s-j)}{(c-i+1)(j+1)} & \text{for } i \geq 1 \text{ and } j \leq s-1, \\
\beta(i-1,j) & = & \beta(i,j)\frac{i(1-\mu)}{(c-i+1)\mu} & \text{for } i \geq 1, \\
\beta(i+1,j) & = & \beta(i,j)\frac{(c-i)\mu}{(i+1)(1-\mu)} & \text{for } i \leq c-1, \\
\beta(i+1,j-1) & = & \beta(i,j)\frac{(c-i)j}{(i+1)(s-j+1)} & \text{for } i \leq c-1 \text{ and } j \geq 1.
\end{array}$$

For the first protocol, the transition probability matrix P is given by relations (11). Using these relations and relations above, we obtain for $i = 1, \ldots, c-1$ and $j = 1, \ldots, s-1$,

$$\begin{aligned}
(\beta P)(i,j) &= \beta(i,j)p_{(i,j),(i,j)} + \beta(i,j+1)p_{(i,j+1),(i,j)} + \beta(i,j-1)p_{(i,j-1),(i,j)} \\
&\quad + \beta(i-1,j+1)p_{(i-1,j+1),(i,j)} + \beta(i-1,j)p_{(i-1,j),(i,j)} \\
&\quad + \beta(i+1,j)p_{(i+1,j),(i,j)} + \beta(i+1,j-1)p_{(i,j),(i,j)} \\
&= \beta(i,j)\left(\frac{ij\mu + (c-i)(s-j)(1-\mu)}{(c+s)s} + \frac{j\mu + (s-j)(1-\mu)}{c+s}\right) \\
&\quad + \beta(i,j)\frac{\mu(s-j)(i+s)}{(c+s)s} + \beta(i,j)\frac{(1-\mu)j(c-i+s)}{(c+s)s} \\
&\quad + \beta(i,j)\frac{i(s-j)(1-\mu)}{(c+s)s} + \beta(i,j)\frac{ij(1-\mu)}{(c+s)s} \\
&\quad + \beta(i,j)\frac{(c-i)(s-j)\mu}{(c+s)s} + \beta(i,j)\frac{(c-i)j\mu}{(c+s)s} \\
&= \beta(i,j)\Big(\frac{ij\mu + (c-i)(s-j)(1-\mu)}{(c+s)s} + \frac{j\mu + (s-j)(1-\mu)}{c+s} + \frac{\mu(s-j)i}{(c+s)s} \\
&\quad + \frac{\mu(s-j)}{c+s} + \frac{(1-\mu)j(c-i)}{(c+s)s} + \frac{(1-\mu)j}{c+s} + \frac{i(1-\mu)}{c+s} + \frac{(c-i)\mu}{c+s}\Big) \\
&= \beta(i,j).
\end{aligned}$$

When $i = 0$ or $i = c$ and $j = 0$ or $j = s$ we easily obtain the same result.

For the second protocol, the transition probability matrix P is given by relations (12). For $i = 1, \ldots, c-1$ and $j = 1, \ldots, s-1$, we have

$$\begin{aligned}(\beta P)(i,j) &= \beta(i,j)\frac{j\mu + (s-j)(1-\mu)}{c+s} + \beta(i,j+1)\frac{(j+1)(1-\mu)}{c+s} \\ &+ \beta(i,j-1)\frac{(s-j+1)\mu}{c+s} + \sum_{(u,v)\in S_{i+j+1}} \beta(u,v)\frac{u(1-\mu)}{c+s}q(i,i+j) \\ &+ \sum_{(u,v)\in S_{i+j}} \beta(u,v)\left(\frac{u\mu}{c+s}q(i,i+j-1) + \frac{(c-u)(1-\mu)}{c+s}q(i,i+j)\right) \\ &+ \sum_{(u,v)\in S_{i+j-1}} \beta(u,v)\frac{(c-u)\mu}{c+s}q(i,i+j-1),\end{aligned}$$

where S_ℓ is the set defined by $S_\ell = \{(u,v) \mid 0 \le u \le c,\ 0 \le v \le c \text{ and } u+v=\ell\}$. Using the recurrence relations above on β and two variables changes $u := u+1$ and $u := u-1$, we obtain

$$\begin{aligned}(\beta P)(i,j) &= \beta(i,j)\left(\frac{j\mu + (s-j)(1-\mu)}{c+s} + \frac{(s-j)\mu}{c+s} + \frac{j(1-\mu)}{c+s}\right) \\ &+ \sum_{(u,v)\in S_{i+j}} \beta(u+1,v)\frac{(u+1)(1-\mu)}{c+s}q(i,i+j) \\ &+ \sum_{(u,v)\in S_{i+j}} \beta(u,v)\left(\frac{u\mu}{c+s}q(i,i+j-1) + \frac{(c-u)(1-\mu)}{c+s}q(i,i+j)\right) \\ &+ \sum_{(u,v)\in S_{i+j}} \beta(u-1,v)\frac{(c-u+1)\mu}{c+s}q(i,i+j-1) \\ &= \beta(i,j)\frac{s}{c+s} + \sum_{(u,v)\in S_{i+j}} \beta(u,v)\frac{(c-u)\mu}{c+s}q(i,i+j) \\ &+ \sum_{(u,v)\in S_{i+j}} \beta(u,v)\left(\frac{u\mu}{c+s}q(i,i+j-1) + \frac{(c-u)(1-\mu)}{c+s}q(i,i+j)\right) \\ &+ \sum_{(u,v)\in S_{i+j}} \beta(u,v)\frac{u(1-\mu)}{c+s}q(i,i+j-1),\end{aligned}$$

which leads to

$$(\beta P)(i,j) = \frac{\beta(i,j)s}{c+s} + \sum_{(u,v)\in S_{i+j}} \beta(u,v)\left(\frac{c-u}{c+s}q(i,i+j) + \frac{u}{c+s}q(i,i+j-1)\right).$$

By definition of $q(i,i+j)$, we have

$$q(i,i+j-1) = q(i,i+j)\frac{j(c+s-(i+j))}{(i+j)(s-j)}$$

and by definition of $\beta(u,v)$, we have

$$\sum_{(u,v)\in S_{i+j}} u\beta(u,v) = \binom{c+s}{i+j}\mu^{i+j}(1-\mu)^{c+s-(i+j)}\frac{(i+j)c}{c+s},$$

and thus

$$\sum_{(u,v)\in S_{i+j}} (c-u)\beta(u,v) = \binom{c+s}{i+j}\mu^{i+j}(1-\mu)^{c+s-(i+j)}\frac{c(c+s-(i+j))}{c+s}.$$

This leads to

$$(\beta P)(i,j) = \frac{\beta(i,j)s}{c+s} + \frac{\binom{c+s}{i+j}\mu^{i+j}(1-\mu)^{c+s-(i+j)}q(i,i+j)cs(c+s-(i+j))}{(c+s)^2(s-j)}.$$

Again, by definition of $q(i, i+j)$, we have

$$\binom{c+s}{i+j}\mu^{i+j}(1-\mu)^{c+s-(i+j)}q(i,i+j) = \beta(i,j)\frac{(c+s)(s-j)}{s(c+s-(i+j))},$$

which gives $(\beta P)(i,j) = \frac{\beta(i,j)s}{c+s} + \frac{\beta(i,j)c}{c+s} = \beta(i,j)$. As for protocol 1, the result for frontier states is easier to derive.

Theorem **??**

Theorem 1 is quite interesting as it shows that the stationary distribution π is exactly the same for both protocols, and that this distribution is equal to distribution β (independently from the initial distribution). At a first glance, we could guess that this phenomenon is due to the fact that the Markov chain X is the tensor product of two independent Markov chains. However, this is clearly not the case as the behavior of red balls in $\mathcal{C}$ depends on the behavior of red balls in $\mathcal{S}$. This holds for both protocols. The stationary availability of the system defined by the long run probability to be in safe states is denoted by P_{safe} and is given by

$$P_{\text{safe}} = \pi_A \mathbb{1} = \sum_{x=0}^{c'} \binom{c}{x}\mu^x(1-\mu)^{c-x}.$$

This probability can also be interpreted as the long run proportion of time spent in safe states. Note that the stationary distribution does not depend on the size of $\mathcal{S}$.

6 Robustness of the Overlay Network

We now show that by inducing global churn, we can preserve the safety of the system. We consider that we have ℓ identical and independent Markov chains $X^{(1)}, \ldots, X^{(\ell)}$ on the same state space $\Omega = A \cup B$, with initial probability distribution α and transition probability matrix P. Each Markov chain $X^{(i)}$ models a particular cluster of peers and, for $n \geq 0$, N_n represents the number of safe clusters after the n-th round, i.e. the number of Markov chains being in subset A after the n-th transition has been triggered, defined by $N_n = \sum_{j=1}^{\ell} 1_{\{X_n^{(j)} \in A\}}$. The ℓ Markov chains being identical and independent, N_n has a binomial distribution, that is, for $k = 0, \ldots, \ell$

$$\begin{aligned}\mathbb{P}\{N_n = k\} &= \binom{\ell}{k}\left(\mathbb{P}\{X_n^{(1)} \in A\}\right)^k\left(1-\mathbb{P}\{X_n^{(1)} \in A\}\right)^{\ell-k} \\ &= \binom{\ell}{k}\left(\alpha P^n \mathbb{1}\right)^k\left(1-\alpha P^n \mathbb{1}\right)^{\ell-k} \qquad (14)\end{aligned}$$

and

$$E(N_n) = \ell \alpha P^n \mathbb{1}. \qquad (15)$$

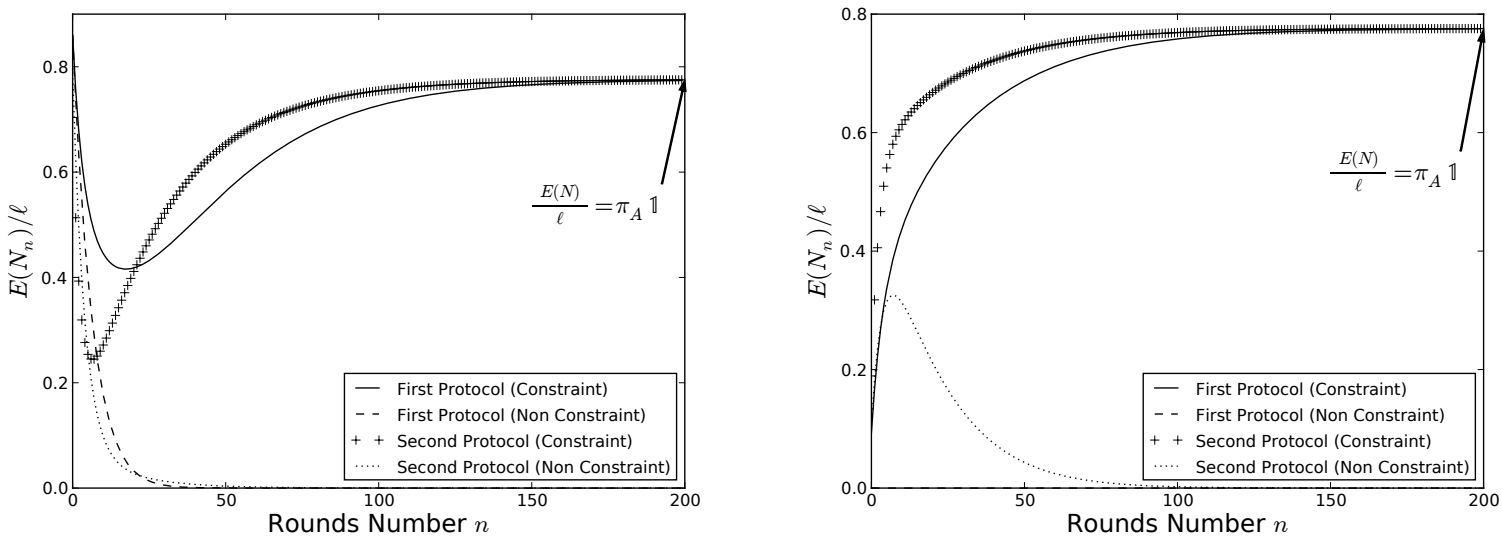
If N denotes the stationary number of safe clusters, we have

$$E(N) = \begin{cases} \ell\pi_A\mathbb{1} & \text{for a constrained adversary} \\ 0 & \text{for a non constrained adversary} \end{cases}$$

These results are illustrated in Figure 11. The main observation is that with a constrained adversary, the expected number of safe clusters for both protocols tends to the same limit $\ell\pi_A\mathbb{1}$ whatever the amount of initial pollution, while with a non constrained adversary all clusters get eventually polluted. This clearly shows that by limiting the time spent by peers at the same position in the overlay targeted attacks are tolerated.

7 Conclusion

In this paper, we have proposed a mechanism that enables the enforcement of limited peers lifetime compliant with DHT-based overlays specificities. We have investigated the long run behavior of several adversarial strategies. Our analysis has demonstrated that an adversary can easily subvert a cluster-based overlay by simply never triggering leave operations. We have shown that when peers have to regularly leave the system, a stationary regime where the ratio of malicious peers is bounded is eventually reached. We are currently implementing the limited peer lifetime mechanism in the cluster-based DHT overlay PeerCube [6]. Preliminary results are encouraging as they show that inducing a moderate churn at all peers of the overlay is sufficient to keep the system safe. Results tend also to show that its management overhead scales logarithmically with the size of the overlay, which makes this solution definitively adapted to large scale and open systems.



(a) $C_r = 3$ and $S_r = 12$, i.e. $X_0 = (3, 12)$

(b) $C_r = 4$ and $S_r = 6$, i.e. $X_0 = (4, 6)$

Figure 11: Percentage of the expected number of safe clusters (relation (15)) as a function of the rounds number n for both protocols, both kind of adversaries and for two different initial states. In these experiments, $\ell = 100$, $c = 10$, $s = 20$, and $\mu = 25\%$.

References

[1] M. Castro, P. Druschel, A. Ganesh, A. Rowstron, and D. S. Wallach, "Secure routing for structured peer-to-peer overlay networks," in *Proceedings of the Symposium on Operating Systems Design and Implementation (OSDI)*, 2002.

[2] A. Singh, T. Ngan, P. Drushel, and D. Wallach, "Eclipse attacks on overlay networks: Threats and defenses," in *Proceedings of the Conference on Computer Communications (INFOCOM)*, 2006.

[3] E. Sit and R. Morris, "Security considerations for peer-to-peer distributed hash tables," in *Proceedings of the International Workshop on Peer-to-Peer Systems (IPTPS)*, 2002.

[4] A. Fiat, J. Saia, and M. Young, "Making chord robust to byzantine attacks," in *Proceedings of the Annual European Symposium on Algorithms (AESA)*, 2005.

[5] I. Baumgart and S. Mies, "S/kademlia: A practicable approach towards secure key-based routing," in *Proceedings of the International Conference on Parallel and Distributed Systems (ICPADS)'*, 2007.

[6] E. Anceaume, F. Brasileiro, R. Ludinard, and A. Ravoaja, "Peercube: an hypercube-based p2p overlay robust against collusion and churn," in *Proceedings of the IEEE International Conference on Self-Adaptive and Self-Organizing Systems (SASO)*, 2008.

[7] M. Srivatsa and L. Liu, "Vulnerabilities and security threats in structured peer-to-peer systems: A quantitiative analysis," in *Proceedings of the 20th Annual Computer Security Applications Conference (ACSAC)*, 2004.

[8] B. Awerbuch and C. Scheideler, "Towards scalable and robust overay networks," in *Proceedings of the International Workshop on Peer-to-Peer Systems (IPTPS)*, 2007.

[9] ——, "Group spreading: A protocol for provably secure distributed name service," in *Proceedings of the 31rst International Colloquium on Automata, Languages and Programming (ICALP)*, 2004.

[10] E. Anceaume, F. Brasileiro, R. Ludinard, B. Sericola, and F. Tronel, "Analytical study of adversarial strategies in cluster-based overlays," in *Proceedings of the 2nd International Workshop on Reliability, Availability, and Security (WRAS)*, 2009.

[11] T. Locher, S. Schmid, and R. Wattenhofer, "equus: A provably robust and locality-aware peer-to-peer system," in *Proceedings of the International Conference on Peer-to-Peer Computing (P2P)*, 2006.

[12] R. Rivest, "Rfc1321: The md5 message-digest algorithm," *Internet Activities Board*, 1992.

[13] C. G. Plaxton, R. Rajaraman, and A. W. Richa, "Accessing nearby copies of replicated objects in a distributed environment," in *Proceedings of the 9th Annual ACM Symposium on Parallel Algorithms and Architectures (SPAA)*, 1997.

[14] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Shenker, "A scalable content-addressable network," in *Proceedings of the ACM SIGCOMM*, 2001.

[15] I. Stoica, D. Liben-Nowell, R. Morris, D. Karger, F. Dabek, M. F. Kaashoek, and H. Balakrishnan, "Chord: A scalable peer-to-peer lookup service for internet applications," in *Proceedings of the ACM SIGCOMM*, 2001.

[16] P. Druschel and A. Rowstron, "Past: A large-scale, persistent peer-to-peer storage utility," in *Proceedings of the 8th Workshop on Hot Topics in Operating Systems (HotOS)*, 2001.

[17] P. Maymounkov and D. Mazieres, "Kademlia: A peer-to-peer information system based on the xor metric," in *Proceedings for the International Workshop on Peer-to-Peer Systems (IPTPS)*, 2002.

[18] L. Lamport, R. Shostak, and M. Pease, "The byzantine generals problem," *ACM Transactions on Programming Languages and Systems*, vol. 4, 1982.

[19] J. Douceur, "The sybil attack," in *Proceedings of the 1st International Workshop on Peer-to-Peer Systems (IPTPS)*, 2002.

[20] D. Dolev, E. Hoch, and R. van Renesse, "Self-stabilizing and byzantine-tolerant overlay network," in *Proceedings of the International Conference On Principles Of Distributed Systems (OPODIS).* LNCS 4878, 2007.

[21] B. Sericola, "Closed form solution for the distribution of the total time spent in a subset of states of a Markov process during a finite observation period," *Journal of Applied Probability*, vol. 27, no. 3, pp. 713–719, 1990.

[22] B. Sericola and G. Rubino, "Sojourn times in Markov processes," *Journal of Applied Probability*, vol. 26, no. 4, pp. 744–756, 1989.