On the impact of pollution attacks on coding-based distributed storage systems

On the impact of pollution attacks on coding-based distributed storage systems

Rossano Gaeta

Abstract—Coding-based distributed storage systems (DSS) are employed in many diverse heterogeneous settings, e.g., cloud storage data centers, peer-to-peer systems, wireless sensor networks, fog/edge computing system, to provide better throughput, latency, reliability, scalability, load adaptation, geographical migration and fault tolerance with respect to traditional monolithic enterprise storage systems. Despite the undoubted advantages offered by coding, reliability and security are jeopardized by a pollution attack that can easily disrupt the entire system and degrade performance.

In this paper we take an abstract view of a DSS and we investigate by means of mathematical modeling what are the availability, robustness, and timeliness of heterogeneous, coding-based DSS when storage nodes (SN) are unreliable and can be malicious. To this end, we focus on a class of allocations of coded fragments to SNs that we call *feasible allocations*; the model takes into account both *reliability* and *reactivity* of SNs.

We define *robust availability* and *timeliness* of feasible allocations that we use to characterize the overall performance and robustness of the DSS in a reference scenario. Our analysis reveals that code redundancy is a double-edged sword in a DSS where malicious SNs come into play and that there exists an optimal value of code redundancy regardless all system parameters that maximizes the number of malicious SNs that can be tolerated to achieve maximum DSS performance. We also found that larger codes are preferred over short ones as they yield superior DSS performance in the presence of malicious SNs. Furthermore, when multiple feasible allocations yield the highest DSS performance timeliness can be used as a guide for the choice. Finally, heterogeneity plays a role in determining the timeliness of the maximally spread allocations in the case of targeted attacks.

Index Terms—distributed storage, cloud storage, coding, security, integrity, heterogenous, pollution attack.

1 Introduction

Distributed storage systems (DSS) are employed in many diverse settings, e.g., cloud storage data centers, peer-to-peer systems, wireless sensor networks, fog/edge computing system, to provide better throughput, latency, reliability, scalability, load adaptation, geographical migration and fault tolerance with respect to traditional monolithic enterprise storage systems [1], [2], [3], [4], [5], [6], [7].

Coding: Often, coding is exploited to improve many aspects of DSSs, i.e., to increase throughput, reduce latency, simplify data collection, and granting reliability [3], [8], [9], [10]. These systems exploit some sort of linear erasure codes (n, k, r) whereby a *storage object* is divided into k source fragments that are used to compute n coded fragments as well as r repair fragments. Then, the coded fragments are spread across the elements of the so called *placement group*, i.e., a subset of the components of the DSS known as the *storage nodes* (SN). Each storage object can be recovered from a subset of $k \leq l \leq n$ coded fragments obtained from the SN of its placement group and a lost coded fragment can be replaced by a new one by using r repair fragments. If the storage object can be recovered from any k of the n coded fragments then an erasure code is termed as a MDS (maximum distance separable) code.

Heterogeneity: Most of the designs and analysis of coding-based DSS assume SNs are homogeneous, e.g., the responding probability to a data collector is the same for all SNs [11]. Nevertheless, virtually all contexts that are amenable to the deployment of DSS are inherently heterogeneous. For instance, peer-to-peer and fog/edge computing systems are by definition composed of terminals with different storage, computing, and communications capabilities, while in cloud storage data centers components with different resource budgets co-exist due to the periodic upgrades and replacements.

Weakness: Despite the undoubted advantages offered by coding, reliability and security of DSS are still concerns that can significantly limit their adoption [12], [13]. In particular, a data modification attack (a.k.a. *pollution attack*) can easily disrupt the entire system and degrade performance since intentional alteration of even a single coded fragment can propagate its bogus effects during the decoding process and damage the original storage object [14].

Our contribution

In this paper we take an abstract view of a DSS and we investigate by means of mathematical modeling what are the availability, robustness, and timeliness of coding-based DSS when SNs can be malicious and could intentionally modify one or more coded fragments representing a storage object. To this end, we focus on a class of allocations of coded fragments to SNs that we call *feasible allocations*. The model takes into account both *reliability*, i.e., the probability a SN responds to a query for its coded fragments, and *reactivity*, i.e., the probability the response from a SN is received by a data collector within a given time deadline. In the model SNs are partitioned into reliability and reactivity *classes* to represent a heterogeneous DSS.

- *Rossano Gaeta is with Università degli Studi di Torino, Dipartimento di Informatica, Torino, Italia. E-mail: rossano.gaeta@unito.it*

We then define two measures for a feasible allocation: the first one we call *robust availability* that we use to analyze the role of code redundancy, i.e., the amount of coded fragments generated to represent a storage object in the DSS, and to investigate the maximum number of malicious SNs that the DSS can tolerate without affecting the possibility of decoding the original storage object and without allowing malicious SNs to collude to recover and modify it. The second one is called *timeliness* of a feasible allocation that we use to characterize the capability of a DSS to provide a data collector with coded fragments representing a storage object within a given time deadline.

We consider a reference scenario and we use the model predictions to provide answers to the following questions:

- what is the role of code redundancy?
- what is the impact of average DSS reliability?
- what is the impact of limiting the size of the placement groups?
- is there an optimal code redundancy?
- which feasible allocations are best for optimal code redundancy?
- what is the role of code and DSS size?
- what is the impact of DSS heterogeneity?
- how does the timeliness of optimal feasible allocations depend on DSS heterogeneity?

Paper organization

The paper is organized as follows: Section 2 describes the coding-based DSS we consider as well as the attack model of malicious SN. Section 3 shows the analytical model for DSS availability, robustness, and overall performance. Section 4 contains the discussion of all values of DSS parameters we used to define the reference scenario. Section 5 comments on the results we obtained for both homogeneous and heterogeneous DSS, while Section 6 discusses the scientific background of our work. Finally, Section 7 summarizes the paper contribution, draws conclusions, and outlines possible future developments of the current research activity.

Notation

To ease the task of the reader Table 1 summarizes the main notation used throughout the paper. All sets have been denoted by calligraphic, upper-case letters.

2 System model

In this section we consider a very abstract DSS. We focus on the description of the key components and functions that can be found and are performed in any kind of DSS. We describe the DSS components and their interactions in Section 2.2 and the attack model in Section 2.3.

2.1 The coding

We assume that storage objects are represented by a set of coded fragments according to a MDS-like coding scheme. In particular, we assume that an (n, k) erasure code with $n \geq k$ is employed such that any subset whose size is k is sufficient for retrieving the original storage object. We also assume that storage objects are encoded by generating redundant coded fragments according to the code redundancy $R = \frac{n}{k}$.

TABLE 1: Paper notation.

Symbol	Description
Coding parameters	
k	Number of source fragments
R	Code redundancy
n	Number of coded fragments
DSS parameters	
M	Size of the DSS
C	Number of reliability classes
n_i	Size of class i SNs
r_i	Reactivity of class i
p_i	Reliability of class i
$\overline{p}$	Average DSS reliability
Allocations	
a_i	Size of the placement group of class i
x_i	Number of coded fragments to each SN in the placement group of class i
$\underline{a} = (a_1, \ldots, a_C)$ $\underline{x} = (x_1, \ldots, x_C)$	Feasible allocation
X_i	Overall number of coded fragments allocated to the placement group of class i
N_S	Overall size of the placement group
$\mathcal{F}(n)$	Set of feasible allocations
Attack model	
m_i	Number of malicious SNs in class i
$\underline{m} = (m_1, \ldots, m_C)$	Feasible attack
N_P	Overall number of malicious SNs
$\mathcal{M}(N_P)$	Set of feasible attacks
$\underline{\mu} = (\mu_1, \ldots, \mu_C)$	Actual attack
$\overline{\mathcal{A}}(\underline{a}, \underline{x}, \underline{m})$	Set of actual attacks
$\underline{\lambda} = (a_1 - \mu_1, \ldots, a_C - \mu_C)$	Honest allocation
$\mathcal{H}(\underline{a}, \underline{x}, \underline{m})$	Set of honest allocations
$\underline{\rho} = (\rho_1, \ldots, \rho_C)$	Honest responding set
$\overline{\mathcal{R}}(\underline{a}, \underline{x}, \underline{m}, \underline{\mu})$	Set of honest responding sets

2.2 The DSS

The DSS model we consider is composed of M SNs and to account for the heterogeneity of SNs we consider them as partitioned in $C > 0$ disjoint subsets called reliability *classes*. Each class i $(1 \leq i \leq C)$ is composed of n_i $(n_i > 0)$ SNs therefore $M = \sum_{i=1}^{C} n_i$. The reliability of SNs in class i is characterized by the parameter $0 < p_i \leq 1$, i.e., p_i is the probability that a SN in class i is successfully accessed to retrieve stored coded fragments. It follows that the DSS is characterized by the *average reliability* $\overline{p} = \frac{\sum_{i=1}^{C} n_i p_i}{M}$. The DSS model also comprises an *allocator* and a *collector*:

- upon *writing* a storage object to the DSS the allocator:
  - encodes it by means of a MDS (n, k) erasure code whose redundancy is R;
  - assigns X_i coded fragments to class i proportionally to the class size, i.e., $X_i = n\frac{n_i}{M}$;
  - selects a random subset of size a_i $(1 \leq a_i \leq n_i)$ from each class i; it follows that the overall size of the placement group is equal to $N_S = \sum_{i=1}^{C} a_i$. The identifiers of the selected placement group are retrieved upon each read/write operation on a storage object and they determine where the coded fragments for that storage object have to be held;
  - assigns x_i coded fragments to each SN of the placement group that belongs to class i such that $X_i = a_i x_i$. This ensures that an equal amount of coded fragments is stored on each

$$h(\underline{a},\underline{x},\underline{m},\underline{\mu}) = \underbrace{\mathbb{1}\left\{\sum_{i=1}^{C} x_i \cdot \mu_i < k\right\}}_{\text{robustness}} \cdot \underbrace{\sum_{\underline{\rho}\in\mathcal{R}(\underline{a},\underline{x},\underline{m},\underline{\mu})} \left[\prod_{i=1}^{C} b(a_i-\mu_i,p_i,\rho_i)\right] \mathbb{1}\left\{\sum_{i=1}^{C} x_i \cdot \rho_i \geq k\right\}}_{\text{availability}}. \tag{1}$$

node of the placement group that belongs to the same reliability class. Furthermore, the selection of random subsets of SN within each class also guarantees that, on average, each SN is included in the same number of placement groups.

We formalize the description of how an allocation is organized by the following:

Definition 2.1. A *feasible allocation* of n coded fragments in a DSS composed of C reliability classes is represented by a pair of vectors of positive integers $(\underline{a},\underline{x})$ where $\underline{a}=(a_1,\ldots,a_C)$ and $\underline{x}=(x_1,\ldots,x_C)$ whose components are such that $\forall i, a_i \leq n_i \wedge x_i \leq n$ and $n=\sum_{i=1}^{C} X_i = \sum_{i=1}^{C} x_i a_i$.

We denote the set of all feasible allocations of n coded fragments in a DSS composed of C reliability classes as $\mathcal{F}(n)$[1].

- upon *reading* a storage object the collector queries the SNs of its placement group to obtain at least k coded fragments to recover it. Recovery of the requested storage object can only occur with some probability because SNs:
  - are assumed to respond to requests only with a given probability, and
  - can be malicious and hence modify coded fragments before responding to the collector.

2.3 The attack model

We consider a system where m_i SNs in reliability class i are *malicious* and intentionally alter coded fragments they store. The collector is thus able to recover the requested storage object only if at least k coded fragments are provided by *honest* SNs. Therefore, we define:

Definition 2.2. A *feasible attack* launched by $N_P \leq M$ malicious SNs is represented by a vector of non-negative integers $\underline{m}=(m_1,\ldots,m_C)$ whose components are such that $\forall i, 0 \leq m_i \leq n_i \wedge N_P = \sum_{i=1}^{C} m_i$.

We denote as $\mathcal{M}(N_P)$ the set of all feasible attacks launched by N_P malicious SNs on a DSS composed of C reliability classes. We also define:

Definition 2.3. An *actual attack* for a feasible allocation $(\underline{a},\underline{x})$ and a feasible attack $\underline{m}$ is represented by a vector of non negative integers $\underline{\mu}=(\mu_1,\ldots,\mu_C)$ whose components are such that $\forall i, 0 \leq \mu_i \leq \min(a_i,m_i)$.

Definition 2.4. For a given actual attack $\underline{\mu}$ its corresponding *honest allocation* is represented by the complement vector of non negative integers $\underline{\lambda}=(a_1-\mu_1,\ldots,a_C-\mu_C)$. Furthermore, an *honest responding set* is represented by a vector of non negative integers $\underline{\rho}=(\rho_1,\ldots,\rho_C)$ whose components are such that $\forall i, 0 \leq \rho_i \leq a_i-\mu_i$.

We denote as $\mathcal{A}(\underline{a},\underline{x},\underline{m})$ the set of all actual attacks that can realize for feasible allocation $(\underline{a},\underline{x})$ under feasible attack $\underline{m}$ and as $\mathcal{H}(\underline{a},\underline{x},\underline{m})$ the set of all honest allocations. Finally, for an actual attack $\underline{\mu}\in\mathcal{A}(\underline{a},\underline{x},\underline{m})$ we denote as $\mathcal{R}(\underline{a},\underline{x},\underline{m},\underline{\mu})$ the set of all honest responding sets.

3 The mathematical model

In this section we define the concepts of *robust availability* and *timeliness* of a feasible allocation. Robust availability characterizes the ability to recover a storage object when SNs can be unreliable while minimizing the possibility that malicious SNs collude to recover a storage object and forge a fake one. Timeliness is the probability the data collector recovers a storage object within a pre-specified time deadline. Based on these indicators we define the concept of optimal allocations of coded fragments representing a storage object.

3.1 Preliminaries

If we consider a feasible allocation $(\underline{a},\underline{x})$, a feasible attack $\underline{m}$, and an actual attack $\underline{\mu}$ then we can express the probability that μ_i SNs in class i are malicious as

$$g(n_i,m_i,a_i,\mu_i) = \frac{\binom{m_i}{\mu_i}\binom{n_i-m_i}{a_i-\mu_i}}{\binom{n_i}{a_i}},$$

that is, the hyper-geometric distribution with parameters n_i, m_i, and a_i. We also assume that allocations in different classes are independent therefore the realization probability of an actual attack scenario $\underline{\mu}$ is given by

$$G(\underline{a},\underline{x},\underline{m},\underline{\mu}) = \prod_{i=1}^{C} g(n_i,m_i,a_i,\mu_i).$$

Furthermore, we use the symbol $\mathbb{1}\{\}$ for the *indicator function* that is equal to 1 if the argument is a true statement and 0 otherwise.

3.2 Robust availability of a feasible allocation

To characterize the robust availability of a feasible allocation $(\underline{a},\underline{x})$ under a feasible attack $\underline{m}$, we consider an actual attack $\underline{\mu}\in\mathcal{A}(\underline{a},\underline{x},\underline{m})$ and a honest responding set $\underline{\rho}\in\mathcal{R}(\underline{a},\underline{x},\underline{m},\underline{\mu})$. The probability ρ_i honest SNs respond to the data collector follows a binomial probability distribution whose parameters are $a_i-\mu_i$ and p_i that we denote as $b(a_i-\mu_i,p_i,\rho_i)$. Thanks to the independence assumptions we made, the probability that honest responding sets $\underline{\rho}\in\mathcal{R}(\underline{a},\underline{x},\underline{m},\underline{\mu})$ provide at least k clean coded fragments to the data collector and malicious SNs are not able to forge a fake storage object is given by Equation 1. Then, the probability to recover

[1] Henceforth we drop the explicit dependence on the number of classes C to avoid cluttering the notation.

a clean storage object for feasible allocation $(\underline{a}, \underline{x})$ under feasible attack $\underline{m}$ while avoiding fatal collusion of malicious SNs is given by

$$S(\underline{a}, \underline{x}, \underline{m}) = \sum_{\underline{\mu} \in \mathcal{A}(\underline{a}, \underline{x}, \underline{m})} G(\underline{a}, \underline{x}, \underline{m}, \underline{\mu}) h(\underline{a}, \underline{x}, \underline{m}, \underline{\mu}). \tag{2}$$

Finally, the robust availability of a feasible allocation $(\underline{a}, \underline{x})$ when attackers take control of N_P out of N SNs can be defined as:

$$A(N_P, \underline{a}, \underline{x}) = \frac{\sum\limits_{\underline{m} \in \mathcal{M}(N_P)} S(\underline{a}, \underline{x}, \underline{m})}{|\mathcal{M}(N_P)|}. \tag{3}$$

Robust availability represents the average value of the probability malicious SNs cannot collude to alter a storage object and it is possible to recover a clean storage object for the feasible allocation $(\underline{a}, \underline{x})$ over all possible feasible attacks of N_P malicious SNs.

3.3 Timeliness of a feasible allocation

We consider a feasible allocation $(\underline{a}, \underline{x})$, a feasible attack $\underline{m}$, and an actual attack $\underline{\mu}$. We further assume that:

- the data collector has a time deadline t_d to be met before recovering the original storage object, and
- the probability a response from a SN in class i is received within t_d time units when a response is provided is equal to r_i. We call r_i the *reactivity* of SNs in class i.

In this case, simple theory of order statistics [15] yields the probability that j_i honest SNs in placement group i provide a response within t_d time units as $b(a_i - \mu_i, p_i r_i, j_i)$, i.e., a binomial probability distribution with parameters $a_i - \mu_i$ and $p_i r_i$. It then follows that the probability the data collector receives at least k clean coded fragments within t_d time units (that we denote as $T(N_P, \underline{a}, \underline{x})$) can be obtained from availability $A(N_P, \underline{a}, \underline{x})$ by considering $p_i r_i$ instead of p_i in the derivation of Equation 1.

3.4 DSS overall performance

We define the concept of overall *performance* of the DSS as the highest value of the robust availability over all possible feasible allocations, i.e.,

$$P(n, N_P) = \max_{(\underline{a}, \underline{x}) \in \mathcal{F}(n)} A(N_P, \underline{a}, \underline{x}), \tag{4}$$

along with the set of feasible allocations yielding $P(n, N_P)$ that is given by

$$\mathcal{P}(n, N_P) = \arg\max_{(\underline{a}, \underline{x}) \in \mathcal{F}(n)} A(N_P, \underline{a}, \underline{x}). \tag{5}$$

The optimal size of the placement group when N_P malicious SNs are part of the DSS is given by

$$N_S^*(n, N_P) = \min_{(\underline{a}, \underline{x}) \in \mathcal{P}(n, N_P)} \sum_{i=1}^{C} a_i. \tag{6}$$

Finally, also consider the maximum tolerable attack load as

$$N_P^*(n) = \max_{N_P} \mathbb{1} \left\{ P(n, N_P) \geq 1 - \epsilon \right\}. \tag{7}$$

i.e., the maximum value of N_P that yields the DSS highest performance. In all our experiments we used $\epsilon = 10^{-9}$.

TABLE 2: Reference scenario.

Symbol	Description
Coding parameters	
k	$\{8, 16, 64\}$
R	$\{1.5, 2.0, 2.5, 3.0, 3.5\}$
DSS parameters	
M	$\{128, 512, 1024\}$
C	$\{1, 2, 3\}$
$\overline{p}$	$\{0.7, 0.8, 0.9, 1\}$
Attack model	
N_P	$[0, \frac{M}{2}]$

TABLE 3: Characteristics or reliability classes.

C	Size	Reliability
1	$n_1 = M$	$p_1 = \overline{p}$
2	$n_2 = sM$	$p_2 = 1$
	$n_1 = M - n_2$	$p_1 = \frac{\overline{p} - s}{1 - s}$
3	$n_3 = sM$	$p_3 = 1$
	$n_1 = sM$	$p_1 = 2\overline{p} - 1$
	$n_2 = M - n_1 - n_3$	$p_2 = \overline{p}$

4 Reference scenario

In this section we define the reference scenario we use to evaluate how robust availability and timeliness are affected by system parameters. We identify a few system parameters that impact on the characteristics of the DSS under a pollution attack. Table 2 summarizes all the values of the system parameters we will consider in Section 5 for our analysis.

DSS parameters

The system we consider is composed of M SNs where M ranges in $\{128, 512, 1024\}$. We refer to the cases $M = 128, 512$, and 1024 as the *small, medium,* and *large* size DSS, respectively. We also study the DSS performance for systems hijacked by $N_P = 0, 1, \ldots, \frac{M}{2}$ malicious SNs.

Heterogeneity model

We assume that the M SNs composing the DSS are partitioned in C reliability classes. Without loss of generality, we choose to consider class 1 as the less reliable (we name it the *unreliable class*) and class C as the most reliable (we name it the *reliable class*). Table 3 summarizes the characteristics of reliability classes for the values of C we consider. For the heterogeneous case, i.e., $C > 1$, we assume the size of the reliable class is a fraction s of the size of the DSS (in our experiments we set $s = \frac{1}{8}$. Furthermore, we assume the reliability of the reliable class is always $p_C = 1$ while the reliability of other classes are a function of the desired average reliability of the DSS $\overline{p}$.

Coding parameters

In all our experiments we consider an allocator that uses a (n, k) MDS erasure code to encode a storage object. We consider k ranging in $\{8, 16, 64\}$. We refer to the cases $k = 8, 16$, and 1024 as the *small, medium,* and *large* codes, respectively. For each value of k we study the system characteristics for coding redundancy $R \in [1.5, 3.5]$.

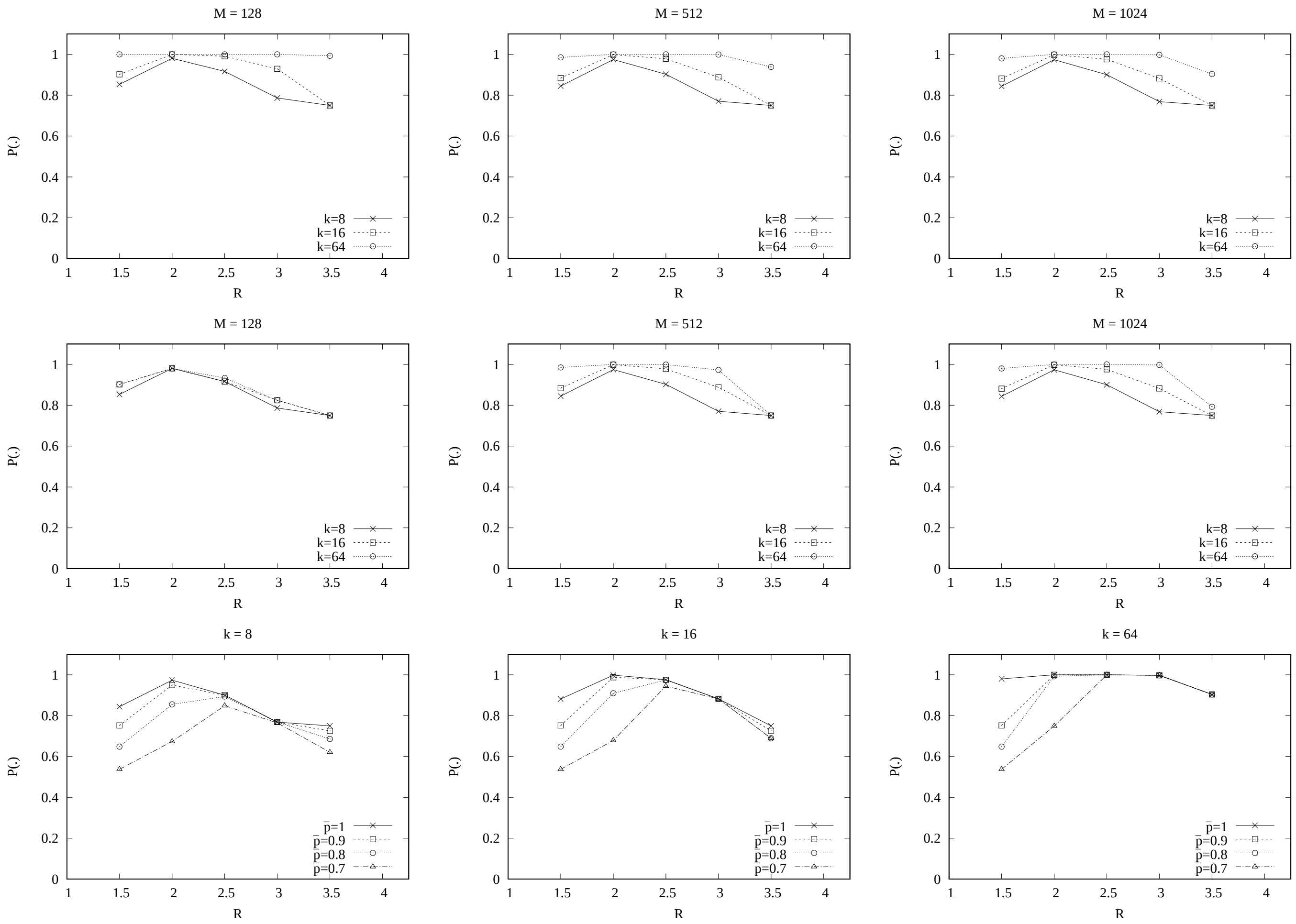

Fig. 1: DSS overall performance $P(.)$ of a homogeneous ($C = 1$), fully reliable ($\overline{p} = 1$) DSS when $N_P = \frac{M}{4}$ as a function of code redundancy R (top row), when the maximum size of the placement group is limited to $\frac{M}{5}$ (middle row), and for a large size ($M = 1024$), unreliable DSS (bottom row).

5 Results

In this section we exploit the model we developed and the performance indexes we defined in Section 3 to assess the overall performance of the DSS we consider. Section 5.1 discusses the impact of system parameters on a homogeneous DSS where only one class of SNs exists, i.e., $C = 1$, while Section 5.2 deals with a heterogeneous DSS where $C > 1$. The discussion of results will be organized as a series of question and answers and to avoid cluttering the notation and the graphs we drop the explicit dependence on all parameters for indexes defined in Section 3.

5.1 Homogeneous DSS

In this section we consider a homogeneous DSS, i.e. $C = 1$, whereby all SNs share the same reliability characteristics. The questions we provide an answer to are:

- ***What is the role of code redundancy*** R***?***
  To answer this question we first consider a fully reliable ($\overline{p} = 1$) DSS whereby the number of malicious SNs is $N_P = \frac{M}{4}$. Figure 1 (top row) shows DSS overall performance $P(.)$ as a function of code redundancy for a small (left graph), medium (middle graph), and large (right graph) size DSS and for all code sizes k.

We note that regardless both code size k and DSS size M, the overall performance $P(.)$ *is not* a monotonic function of R since while availability is an increasing function of R robustness to collusion is a decreasing function of R, instead. More precisely, this can be explained by noting that for a fixed number of malicious SNs (N_P):

- the availability factor in the definition of probability in Equation 1 is proportional to code redundancy R through quantities x_i, i.e., the higher R the more (and the higher) the possible values of x_i;
- the robustness factor is inversely proportional to R through quantities x_i therefore making robustness a decreasing function of R.

The non-monotonicity of $P(.)$ is an intrinsic characteristic regardless all other system parameters. Indeed, Figure 1 (middle row) shows the same results in the case the size of the placement group is limited to $\frac{M}{5}$ while Figure 1 (bottom row) shows the DSS performance when SNs are not fully reliable: although

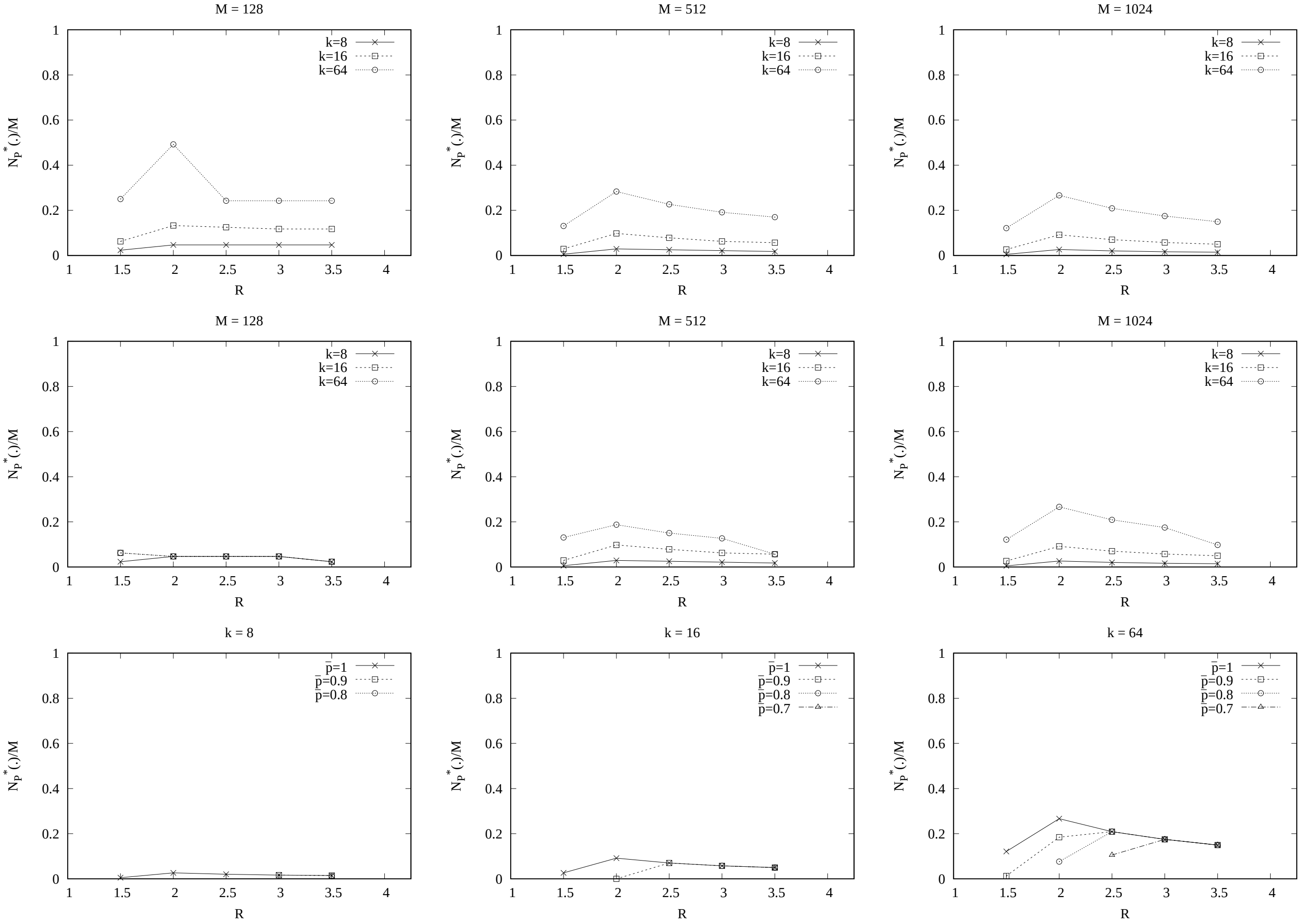

Fig. 2: DSS normalized maximum tolerable attack load $\frac{N_P^*(.)}{M}$ of a homogeneous ($C = 1$), fully reliable ($\overline{p} = 1$) DSS when $N_P = \frac{M}{4}$ as a function of code redundancy R (top row), when the maximum size of the placement group is limited to $\frac{M}{5}$ (middle row), and for a large size ($M = 1024$), unreliable DSS (bottom row).

quantitatively different, the results show the same qualitative behavior. Finally, we also observe that in all cases we considered non-monotonicity of $P(.)$ appears for any code and DSS size.

Conclusions: Code redundancy R is a double-edged sword in a DSS where malicious SNs come into play and a trade-off must be made between availability and robustness to collusion.

- ***Is there an optimal code redundancy R^*?***
  Previous analysis suggests that optimal values for code redundancy R^* can be identified to ensure both maximum availability and robustness to collusion. To this end, in Figure 2 (top row) we show results for the normalized maximum tolerable attack load $\frac{N_P^*(.)}{M}$ as a function of code redundancy R for a fully reliable DSS. Results clearly show that there exists an optimal value for code redundancy among those we considered, i.e., $R^* = 2$, guaranteeing both maximum availability and robustness to collusion for all combinations of code size k and DSS size M. Again, the same conclusions can be drawn in the case the size of the placement group is limited to $\frac{M}{5}$ (Figure 2 middle row) and for unreliable SNs (Figure 2 bottom row).

  Conclusions: In the design of a DSS there exists an optimal value of code redundancy regardless all system parameters, i.e., $R^* = 2$, that maximizes the number of malicious SNs that can be tolerated to achieve maximum DSS performance.

- ***What is the role of code size k and DSS size M?***
  The analysis of results we presented in Figures 1 and 2 also suggests a role for the code size k since it can be noted that the higher k the higher both $P(.)$ and $\frac{N_P^*(.)}{M}$. In particular, the DSS normalized maximum tolerable attack load $\frac{N_P^*(.)}{M}$ tops 0.5 in the case of small DSS and $k = \frac{M}{2}$, i.e., for $M = 128$ and $k = 64$ as depicted in Figure 2 by the top row leftmost graph. To verify this limit in Figure 3 we show the DSS normalized maximum tolerable attack load $\frac{N_P^*(.)}{M}$ of a homogeneous ($C = 1$), fully reliable ($\overline{p} = 1$) DSS as a function of code redundancy R for large M and k. It can be noted that regardless the DSS size M the maximum performance is reached when the code size is $k = \frac{M}{2}$ for code redundancy $R^* = 2$. Higher values of k do not improve the DSS performance.

  Conclusions: Larger codes are preferred over short

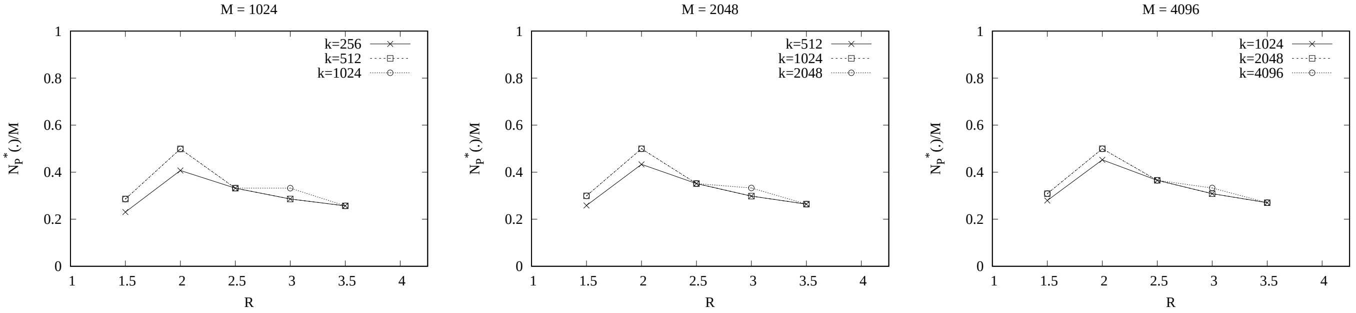

Fig. 3: DSS normalized maximum tolerable attack load $\frac{N_P^*(.)}{M}$ of a homogeneous ($C = 1$), fully reliable ($\overline{p} = 1$) DSS as a function of code redundancy R for large M and k.

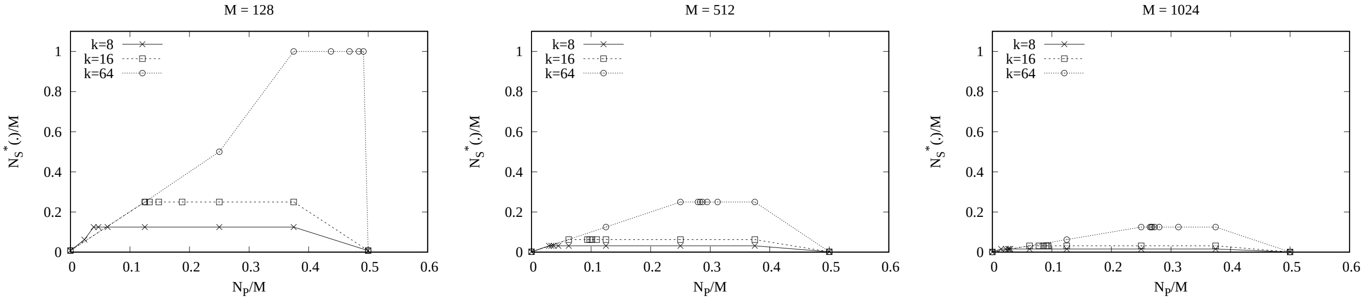

Fig. 4: Normalized optimal size of the placement group $\frac{N_S^*(.)}{M}$ of a homogeneous ($C = 1$), fully reliable ($\overline{p} = 1$) DSS for optimal redundancy $R^* = 2$ as a function of normalized attack load $\frac{N_P}{M}$.

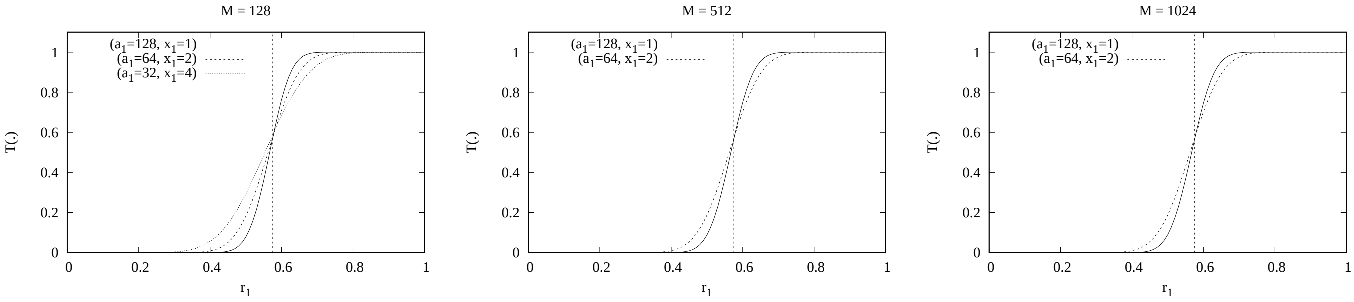

Fig. 5: Timeliness $T(.)$ of equally robust allocations in $\mathcal{P}(.)$ for $R^* = 2$, $k = 64$, and $N_P = \frac{M}{8}$ in the case small (left graph), medium (middle graph), and large (right graph) size DSS as a function of reactivity r_1.

ones as they yield superior DSS performance in the presence of malicious SNs. In particular, code size $k = \frac{M}{2}$ for code redundancy $R^* = 2$ is the minimum value to obtain the highest performance for the DSS for both robust availability and normalized maximum tolerable attack load.

- ***Which feasible allocations are best for optimal code redundancy R^*?***
  Results shown in Figure 4 provide information on the optimal allocations of coded fragments when storage object are stored with redundancy R^*. The graphs show the normalized optimal size of the placement group $\frac{N_S^*(.)}{M}$ of a homogeneous ($C = 1$), fully reliable ($\overline{p} = 1$) DSS for optimal redundancy $R^* = 2$ as a function of normalized attack load $\frac{N_P}{M}$. First of all we note that all curves must have an upper bound at $\min(1, \frac{R^* \cdot k}{M})$ that corresponds to allocations whereby the maximum possible number of SNs is used to allocate coded fragments. We also note that maximally spreading coded fragments is the key to obtain the highest values for overall performance $P(.)$, i.e., to resist to attacks from up to $N_P^*(.)$ malicious SNs.
  Results on the normalized optimal size of the placement group $\frac{N_S^*(.)}{M}$ suggest a more detailed analysis of the elements of set $\mathcal{P}(.)$. In Definition 6 we chose to select the most parsimonious allocation in terms of number of involved SNs when $\mathcal{P}(.)$ contains more than one element. Nevertheless, alternative definitions can be sought if timing considerations are taken into account. In particular, we evaluate the timeliness $T(.)$ of feasible allocations in $\mathcal{P}(.)$ in the case $R^* = 2$, $k = 64$, and $N_P = \frac{M}{8}$ as a function of reactivity r_1.

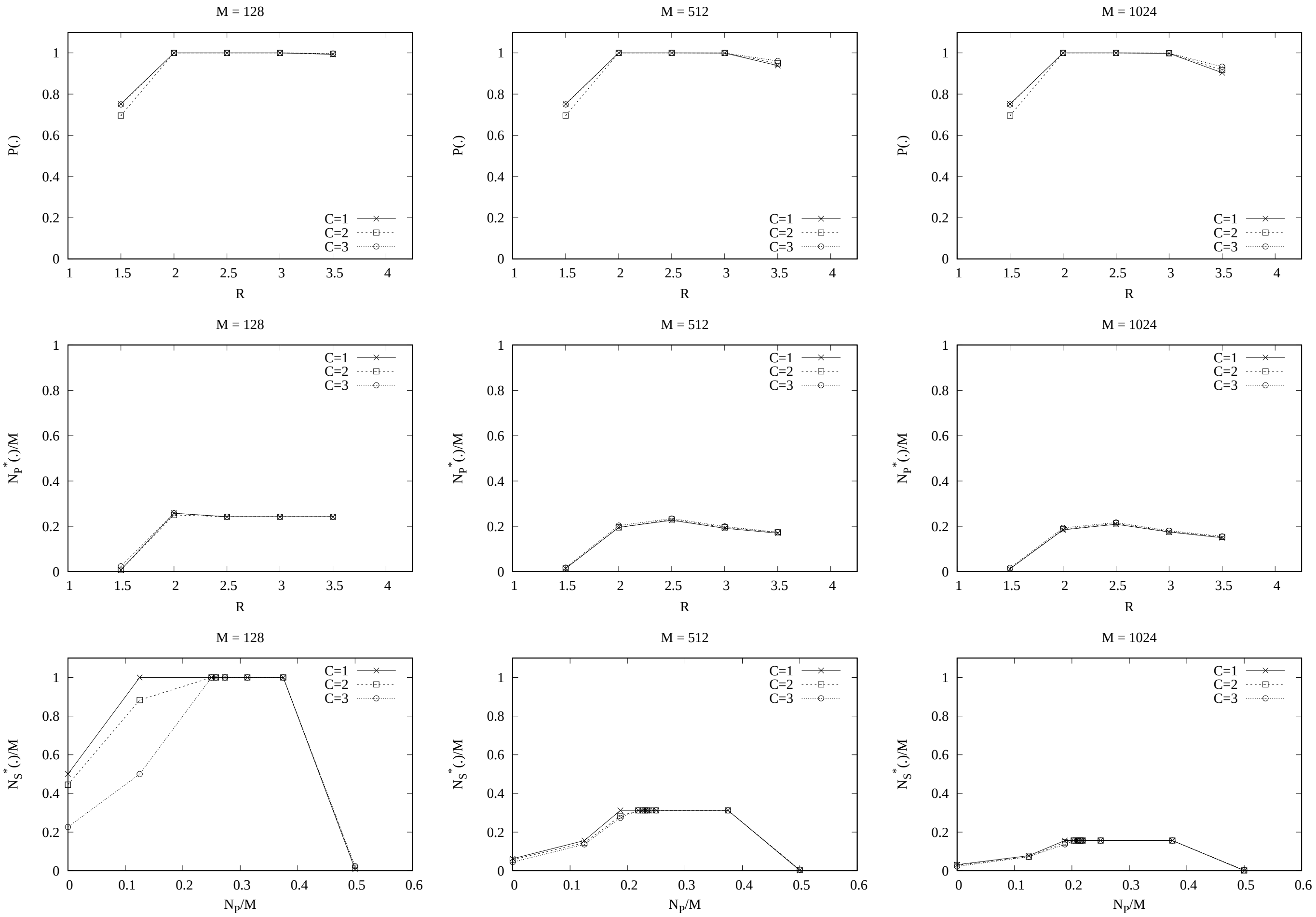

Fig. 6: Results for a heterogeneous, unreliable ($\overline{p} = 0.9$) DSS are displayed for small (left graphs), medium (middle graphs), and large (right graphs) size DSS (M) and for large code size $k = 64$. Top row graphs depict DSS overall performance $P(.)$ when $N_P = \frac{M}{4}$ as a function of code redundancy R, middle row graphs depict DSS normalized maximum tolerable attack load $\frac{N_P^*(.)}{M}$ when $N_P = \frac{M}{4}$ as a function of code redundancy R, while bottom row graphs depict normalized optimal size of the placement group $\frac{N_S^*(.)}{M}$ for optimal redundancy R^* as a function of normalized attack load $\frac{N_P}{M}$.

In Figure 5 we present results for $T(.)$ and we note that in this scenario the set of feasible allocations yielding the highest $P(.)$ always contains two elements, i.e., $\mathcal{P}(.) = \{(128, 1), (64, 2)\}$, but the case $M = 128$ wherein allocation $(32, 4)$ is also possible.

On the one hand, it can be noted that when reactivity is $r_1 < 0.58$ the most parsimonious allocations also maximize timeliness. On the other hand, the maximally spread allocation $(128, 1)$ becomes also optimal for the timeliness when $r_1 \geq 0.58$.

Conclusion: When the number of malicious SNs is fixed and the set of feasible allocations yielding the highest DSS performance $P(.)$ contains more than one element timeliness can be used as a guide for the choice. Indeed, when the probability of SNs responding within a given time deadline is below a system dependent threshold parsimonious allocations are preferred over maximally spread ones while the opposite is true for more reactive responses.

5.2 Heterogeneous DSS

In this section we consider a heterogeneous DSS, i.e. $C > 1$, whereby the average reliability $\overline{p} = 0.9$ and whose reliability classes are defined in Table 3.

- ***What is the impact of DSS heterogeneity C?*** All the observations we made on a homogeneous DSS are valid in the heterogeneous case, as well. Figure 6 shows results for a heterogeneous, unreliable ($\overline{p} = 0.9$) DSS for small (left graphs), medium (middle graphs), and large (right graphs) size DSS (M) and for large code size $k = 64$.

  Top row graphs depict DSS overall performance $P(.)$ when $N_P = \frac{M}{4}$ as a function of code redundancy R: it can be noted that regardless DSS size M, the overall performance $P(.)$ is not a monotonic function of R. Also in the heterogeneous case, non-monotonicity of $P(.)$ is an intrinsic characteristic regardless all other system parameters therefore to avoid cluttering the presentation we omit the graphs showing results in the case the size of the placement group is limited to

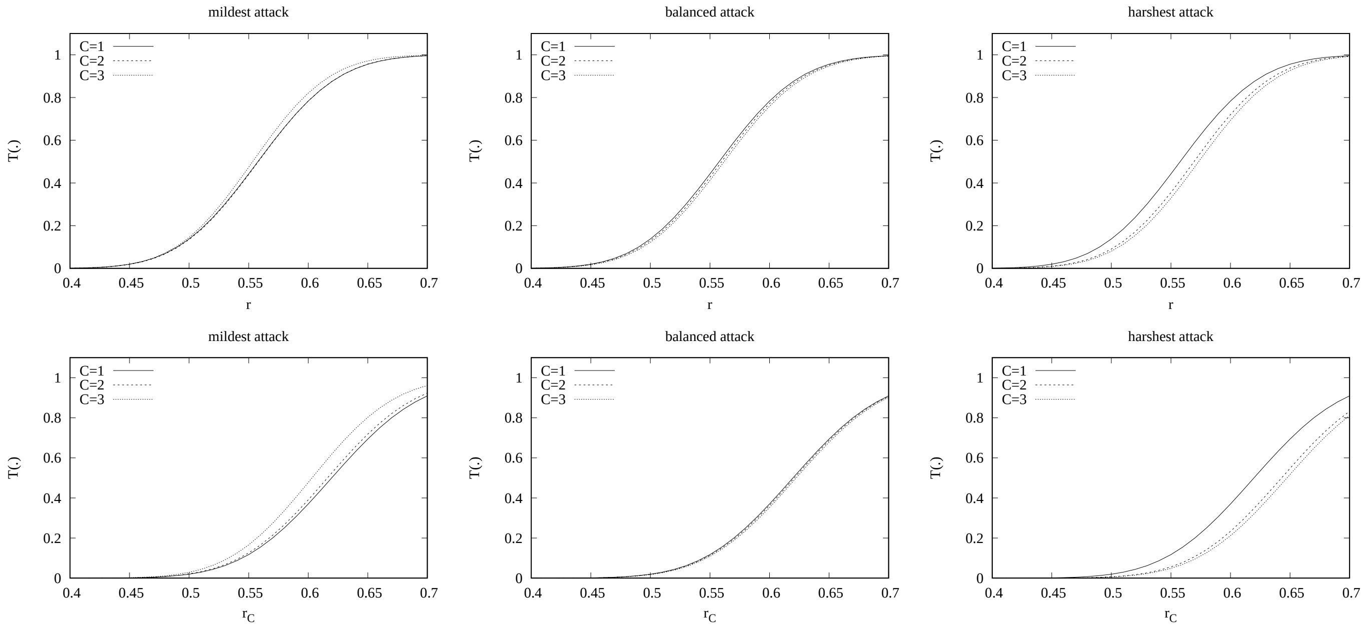

Fig. 7: Timeliness $T(.)$ of maximally spread allocations for a large size DSS, large code size, optimal redundancy $R^* = 2.5$, and for the highest tolerable number of malicious SNs $N_P^*(.)$. Results for the mildest attack (left graph), the balanced attack (middle graph), and the harshest attack (right graph) as a function of reactivity homogeneous r (top row) and reliability class dependent reactivity r_C (bottom row).

$\frac{M}{5}$. Heterogeneity plays a small role only for low and high redundancy values R.

Middle row graphs depict DSS normalized maximum tolerable attack load $\frac{N_P^*(.)}{M}$ when $N_P = \frac{M}{4}$ as a function of code redundancy R. Optimal values for code redundancy R^* can be identified but in the heterogenous case we observe that $R^* = 2$ for a small size DSS while $R^* = 2.5$ for medium and large size DSS. The value of R^* is higher in the unreliable case because a higher redundancy is necessary to guarantee the collector receives the amount of coded fragments that are necessary to recover the original storage object. In this case, the number of reliability classes almost has no role provided that the average reliability $\overline{p}$ is constant.

Bottom row graphs depict normalized optimal size of the placement group $\frac{N_S^*(.)}{M}$ for optimal redundancy R^* as a function of normalized attack load $\frac{N_P}{M}$. Also in this case, all curves are upper bounded by $\min(1, \frac{R^* \cdot k}{M})$ and maximally spread allocations still yield the highest DSS performance when $\frac{N_P}{M} = \frac{N_P^*(.)}{M}$. In this case, the number of reliability classes C has an impact when considering a small size DSS. Indeed, the normalized optimal size of the placement group is lower when $C = 3$ when the number of malicious SN is $N_P < \frac{M}{4}$.

Conclusion: All observations we did by analyzing a homogeneous DSS are still valid in the heterogeneous case, i.e., when the number of reliability classes $C > 1$. The higher the number of reliability classes the lower the optimal size of the placement group in a small size DSS when the number of malicious SN is $N_P < \frac{M}{4}$.

- ***What is the impact of DSS heterogeneity C on timeliness of maximally spread allocations?***
  As we discussed, maximally spread allocations are those that allow the DSS to resist to a pollution attack brought by the highest number of colluding malicious SNs. To analyze the impact of heterogeneity on the timeliness of maximally spread allocations we consider a large size DSS, large code size, optimal redundancy $R^* = 2.5$, and for $N_P = N_P^*(.)$. We also focus on three cases for the pollution attack (two of them are extreme cases):
  - the *mildest* attack that occurs when all SNs in the unreliable class (class 1) are malicious and $n_2 - N_P^*(.) + n_1$ SNs in class 2 are too,
  - the *harshest* attack that occurs when all SNs in the reliable class (class C) are malicious and $n_{C-1} - N_P^*(.) + n_C$ SNs in class $C-1$ are too, and
  - the *balanced* attack that occurs when $m_i = N_P^*(.) \frac{n_i}{M}$.

  Figure 7 (top row) shows results for the mildest attack (left graph), the balanced attack (middle graph), and the harshest attack (right graph) as a function of reactivity r (here we consider all SNs having the same reactivity r regardless the reliability class they belong to). It can be noted that timeliness of the maximally spread allocation lowers as the number of reliability classes C increases in both the balanced and the harshest attack. In the case of the mildest attack high heterogeneity yields a slightly more reactive DSS.

Figure 7 (bottom row) also shows timeliness of maximally spread allocations in the case reactivity of class C is an independent parameter r_C and reactivity of other reliability classes are lower, i.e., reactivity of reliability class i is set to $r_i = p_i r_C$. In this scenario the same observations can be made and the gap among timelinesses for different heterogeneous level are larger.

Conclusion: Heterogeneity plays a role in determining the timeliness of the maximally spread allocations, i.e., the allocations that allow the DSS to resist to a pollution attack launched by $N_P^*(.)$ malicious SNs. In particular, the higher the number of reliability classes C the worst the responsiveness of the allocations in the case of a targeted attack to most reliable SNs, i.e., those in class C.

6 Related works

Several papers dealt with the problem of finding allocations in DSSs with fixed code redundancy where either reliabilities or capacities of SN are not homogeneous, e.g., [16], [17], [18], [19], [20], [21], [22]. All these papers seek to find optimal allocations but do not consider attacks to the DSSs brought by possibly colluding malicious SNs.

Another line of research considers malicious SNs in DSS that is assumed to always guarantee a certain desired level of reliability. In these systems SNs are not reliable but when a SN fails it is replaced by a new node with the same storage capacity. A repair mechanism is then sought and the focus is on determining the system capacity for different type of attacks in both homogenous [23], [24] and heterogeneous capacities DSS [25]. In this setting optimal erasure codes [26] have been devised that meet the lower bounds on the amount of storage and network requirements established in [23]. Also, the secrecy capacity of Minimum Storage Regenerating Codes is investigated in [27].

7 Conclusions and Future Developments

In this paper we focused on coding-based, heterogeneous DSS whereby SNs can be unreliable and malicious. In particular, we considered the so called pollution attack whose impact can be devastating on the performance, reliability, and security of DSS. We took an abstract view of a DSS and we developed a mathematical model to represent the main characteristics of a DSS whose SNs are characterized by a certain level of reliability and reactivity. We considered a special class of allocations of coded fragments to SNs that we named *feasible allocations*. We then used the model to define two measures for a feasible allocation: *robust availability* and *timeliness*. Starting from the indicators we characterized the overall performance and robustness of the DSS in a reference scenario. Our analysis revealed that code redundancy is a double-edged sword in a DSS where malicious SNs come into play, there exists an optimal value of code redundancy regardless all system parameters that maximizes the number of malicious SNs that can be tolerated to achieve maximum DSS performance, larger codes are preferred over short ones as they yield superior DSS performance in the presence of malicious SNs, when multiple feasible allocations yield the highest DSS performance timeliness can be used as a guide for the choice, and heterogeneity plays a role in determining the timeliness of the maximally spread allocations in the case of targeted attacks.

In the current work we considered erasure codes because of their simplicity and their wide adoption in DSSs. We basically considered a system where coded fragments are all functionally equivalent. This means it is possible to focus only on the *number* of source (k) and coded (n) fragments. Furthermore, by assuming a MDS-like coding scheme a further simplification is possible since successful reading of the original storage object can be considered whenever any subset of k fragments are retrieved by the data collector. Nevertheless, some of the assumptions could be relaxed to represent other coding mechanisms. For instance, we could consider modeling of coding algorithms where coded fragments are not all functionally equivalent. In this case, the model could be extended with a reasonable effort to deal with different *functional classes* of coded fragments that would add to the partition of storage nodes into *reliability classes*.

Another interesting topic to investigate could be the analysis of robustness of coding mechanisms such as convolutional codes to intentional modification of coded data.

Finally, we are currently working on the definition of malicious SNs identification techniques in the same line as [28], [29], [30].

References

[1] S. Ghemawat, H. Gobioff, and S.-T. Leung, “The Google File System,” in *ACM SOSP*, Oct. 2003.

[2] K. Shvachko, H. Kuang, S. Radia, and R. Chansler, “The hadoop distributed file system,” in *Mass storage systems and technologies (MSST), 2010 IEEE 26th symposium on*. Ieee, 2010, pp. 1–10.

[3] A. G. Dimakis, K. Ramchandran, Y. Wu, and C. Suh, “A survey on network codes for distributed storage,” *Proceedings of the IEEE*, vol. 99, no. 3, pp. 476–489, 2011.

[4] C. Huang, H. Simitci, Y. Xu, A. Ogus, B. Calder, P. Gopalan, J. Li, S. Yekhanin *et al.*, “Erasure coding in windows azure storage.” in *Usenix annual technical conference*. Boston, MA, 2012, pp. 15–26.

[5] S. Muralidhar, W. Lloyd, S. Roy, C. Hill, E. Lin, W. Liu, S. Pan, S. Shankar, V. Sivakumar, L. Tang *et al.*, “f4: Facebook’s warm blob storage system,” in *Proceedings of the 11th USENIX conference on Operating Systems Design and Implementation*, 2014, pp. 383–398.

[6] Y. Xiang, V. Aggarwal, Y. R. Chen, and T. Lan, “Differentiated latency in data center networks with erasure coded files through traffic engineering,” *IEEE Transactions on Cloud Computing*, vol. 7, no. 2, pp. 495–508, 2019.

[7] E. Bacis, S. De Capitani di Vimercati, S. Foresti, S. Paraboschi, M. Rosa, and P. Samarati, “Dynamic allocation for resource protection in decentralized cloud storage,” in *Proc. of the 2019 IEEE Global Communications Conference (GLOBECOM 2019)*, Waikoloa, Hawaii, USA, December 2019.

[8] Q. Liu, D. Feng, H. Jiang, Y. Hu, and T. Jiao, “Systematic erasure codes with optimal repair bandwidth and storage,” *ACM Trans. Storage*, vol. 13, no. 3, 2017.

[9] Q. Liu, D. Feng, Y. Hu, Z. Shi, and M. Fu, “High-performance general functional regenerating codes with near-optimal repair bandwidth,” *ACM Trans. Storage*, vol. 13, no. 2, 2017.

[10] O. Kolosov, G. Yadgar, M. Liram, I. Tamo, and A. Barg, “On fault tolerance, locality, and optimality in locally repairable codes,” *ACM Trans. Storage*, vol. 16, no. 2, 2020.

[11] D. Leong, A. G. Dimakis, and T. Ho, “Distributed storage allocations,” *IEEE Transactions on Information Theory*, vol. 58, no. 7, pp. 4733–4752, 2012.

[12] G. R. Goodson, J. J. Wylie, G. R. Ganger, and M. K. Reiter, “Efficient byzantine-tolerant erasure-coded storage,” in *International Conference on Dependable Systems and Networks, 2004*, 2004, pp. 135–144.

[13] K. K. Rao, J. L. Hafner, and R. A. Golding, "Reliability for networked storage nodes," *IEEE Transactions on Dependable and Secure Computing*, vol. 8, no. 3, pp. 404–418, 2011.

[14] Z. Yu, Y. Wei, B. Ramkumar, and Y. Guan, "An efficient signature-based scheme for securing network coding against pollution attacks," in *INFOCOM 2008. The 27th Conference on Computer Communications. IEEE*, 2008.

[15] H. A. David and H. N. Nagaraja, "Order statistics," *Encyclopedia of Statistical Sciences*, 2004.

[16] V. Ntranos, G. Caire, and A. G. Dimakis, "Allocations for heterogenous distributed storage," in *Information Theory Proceedings (ISIT), 2012 IEEE International Symposium on*. IEEE, 2012, pp. 2761–2765.

[17] G. Xu, S. Lin, G. Wang, X. Liu, K. Shi, and H. Zhang, "Hero: Heterogeneity-aware erasure coded redundancy optimal allocation for reliable storage in distributed networks," in *Performance Computing and Communications Conference (IPCCC), 2012 IEEE 31st International*. IEEE, 2012, pp. 246–255.

[18] Z. Li, T. Ho, D. Leong, and H. Yao, "Distributed storage allocation for heterogeneous systems," in *Communication, Control, and Computing (Allerton), 2013 51st Annual Allerton Conference on*. IEEE, 2013, pp. 320–326.

[19] M. Noori and M. Ardakani, "Allocation for heterogeneous storage nodes," *IEEE Communications Letters*, vol. 19, no. 12, pp. 2102–2105, 2015.

[20] K. P. Roshandeh, M. Noori, M. Ardakani, and C. Tellambura, "Distributed storage allocation for multi-class data," in *2017 IEEE International Symposium on Information Theory (ISIT)*, 2017, pp. 2223–2227.

[21] M. Sardari, R. Restrepo, F. Fekri, and E. Soljanin, "Memory allocation in distributed storage networks," in *2010 IEEE International Symposium on Information Theory*, 2010, pp. 1958–1962.

[22] P. Hu, C. W. Sung, S. W. Ho, and T. H. Chan, "Optimal coding and allocation for perfect secrecy in multiple clouds," *IEEE Trans. on Information Forensics and Security*, vol. 11, no. 2, pp. 388–399, 2016.

[23] S. Pawar, S. El Rouayheb, and K. Ramchandran, "Securing dynamic distributed storage systems against eavesdropping and adversarial attacks," *IEEE Transactions on Information Theory*, vol. 57, no. 10, pp. 6734–6753, 2011.

[24] R. Tandon, S. Amuru, T. C. Clancy, and R. M. Buehrer, "Toward optimal secure distributed storage systems with exact repair," *IEEE Transactions on Information Theory*, vol. 62, no. 6, pp. 3477–3492, 2016.

[25] T. Ernvall, S. El Rouayheb, C. Hollanti, and H. V. Poor, "Capacity and security of heterogeneous distributed storage systems," *IEEE Journal on Selected Areas in Communications*, vol. 31, no. 12, pp. 2701–2709, 2013.

[26] K. V. Rashmi, N. B. Shah, K. Ramchandran, and P. V. Kumar, "Information-theoretically secure erasure codes for distributed storage," *IEEE Transactions on Information Theory*, vol. 64, no. 3, pp. 1621–1646, 2018.

[27] K. Huang, U. Parampalli, and M. Xian, "On secrecy capacity of minimum storage regenerating codes," *IEEE Transactions on Information Theory*, vol. 63, no. 3, pp. 1510–1524, 2017.

[28] R. Gaeta and M. Grangetto, "Identification of malicious nodes in peer-to-peer streaming: A belief propagation based technique," *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, no. 10, pp. 1994–2003, 2013.

[29] L. Buttyan, L. Czap, and I. Vajda, "Detection and recovery from pollution attacks in coding-based distributed storage schemes," *IEEE Transactions on Dependable and Secure Computing*, vol. 8, no. 6, pp. 824–838, 2011.

[30] R. Gaeta and M. Grangetto, "Malicious node identification in coded distributed storage systems under pollution attacks," *ACM Transactions on Modeling and Performance Evaluation of Computing Systems*, vol. 6, no. 3, pp. 1–27, 2021.

Rossano Gaeta Rossano Gaeta received his Laurea and Ph.D. degrees in Computer Science from the University of Torino, Italy, in 1992 and 1997, respectively. He is currently Full Professor at the Computer Science Department, University of Torino. His current research interests include the design and evaluation of coding techniques in distributed storage systems, the modeling of information diffusion in online social networks, and the analysis of popularity and quality impact in recommender systems.