

# Dependability Evaluation of Cooperative Backup Strategies for Mobile Devices<sup>\*</sup>

[Regular Paper]

Ludovic Courtès, Ossama Hamouda,  
Mohamed Kaâniche, Marc-Olivier Killijian, David Powell

*first-name.last-name@laas.fr*  
LAAS-CNRS  
7 avenue du Colonel Roche  
31077 Toulouse cedex 4  
France

## Abstract

*Mobile devices (e.g., laptops, PDAs, cell phones) are increasingly relied on but are used in contexts that put them at risk of physical damage, loss or theft. This paper discusses the dependability evaluation of a cooperative backup service for mobile devices. Participating devices leverage encounters with other devices to temporarily replicate critical data. Permanent backups are created when the participating devices are able to access the fixed infrastructure. Several data replication and scattering strategies are presented, including the use of erasure codes. A methodology to model and evaluate them using Petri nets and Markov chains is described. The paper also discusses the use of erasure codes and the impact of contributor behavior on the cooperative service. We demonstrate that our cooperative backup service decreases the probability of data loss by a factor up to the ad hoc to Internet connectivity ratio.*

## 1. Introduction

Mobile devices (e.g., laptops, PDAs, cell phones) are increasingly relied on but are used in contexts that put them at risk of physical damage, loss or theft. However, fault-tolerance mechanisms available for these devices often suffer from shortcomings. For instance, data “synchronization” mechanisms, which allow one to replicate a mobile device’s data on a desktop machine, usually require that the desktop machine be either physically accessible or reachable *via* the Internet. Use of third-par-

ty backup servers typically also requires access to some network infrastructure.

Unfortunately, in many scenarios where devices are carried along in different places, access to a network infrastructure (e.g., *via* a Wi-Fi access point) is at best *intermittent*. Often, access to a network infrastructure may be too costly and/or inefficient energy-wise and performance-wise to be considered viable “just” for backup. In emergency situations and upon disaster recovery, for instance, infrastructure may well be unavailable for an unspecified amount of time. In such cases, data produced on a mobile device while the network is unreachable cannot be replicated using the aforementioned synchronization techniques and could be lost. Similarly, environments with scarce Internet connectivity, such as those targeted by the “One Laptop per Child” project (OLPC, <http://laptop.org/>), can hardly rely on access to an infrastructure for doing data backup.

We aim to address these issues by providing a *cooperative* backup service, called MoSAIC [4, 15]. Our approach seeks to apply the peer-to-peer data storage and backup paradigm that has been successful on the Internet [7, 16] to data backup among communicating devices. Participating mobile devices share storage resources. When devices encounter, they may exchange critical data and store the received data on behalf of its sender; eventually, when a participating device gains access to the Internet, it forwards data stored on behalf of other nodes to an Internet-accessible store.

The efficiency of this service, from a fault-tolerance viewpoint, depends on a number of environmental factors: it should provide most benefit with a relatively high density of participating devices and intermittent Internet access for all devices, but also needs to account for occa-

---

<sup>\*</sup>This work was partially supported by the MoSAIC project (ACI S&I, French national program for Security and Informatics; see <http://www.laas.fr/mosaic/>), the Hidenets project (EU-IST-FP6-26979), and the ReSIST network (EU-IST-FP6-26764).

sional device failures and potentially malicious contributor behavior. In order to gain confidence about the efficiency of our backup service, we need to assess the impact of these issues on data dependability.

Various replication and data scattering algorithms may be used to implement the cooperative backup service. Replication may be handled by creating full copies of individual data items (we refer to this as *simple replication*) or by more sophisticated *erasure coding* techniques. Choosing between these techniques implies a tradeoff between storage efficiency and data confidentiality [5]. This tradeoff can be done beforehand and is well understood; however, its impact on data dependability, particularly in our scenario, is unclear. The analytical evaluation presented in this paper aims to clarify this.

We analyze the fault-tolerance gain provided by MoSAIC as a function of (i) the various environmental parameters (frequency of Internet access, device encounter rate, node failure rate) and (ii) different replication strategies. Our approach is based on model-based evaluation, which is well suited to support design tradeoff studies and to analyze the impact of several parameters of the design and the environment from the dependability and performance points of view. We expect such an analysis to provide us with better understanding of the dependability gains to be expected from our service.

We identify two main goals. First, the analysis should help us determine under what circumstances MoSAIC is the most beneficial, compared to solutions that do not replicate data in the *ad hoc* domain. Second, it should help us choose among different replication strategies, depending on a given scenario's parameters and user preferences (e.g., target data availability, confidentiality requirements).

The work presented here builds on our previous work on the design of a cooperative backup service for mobile devices [4, 5, 15]. Its major contribution lies in the *dependability evaluation* of data replication and scattering strategies using analytical methods, taking into account a variety of influential parameters. It differs substantially from earlier evaluation work by other authors due to the entirely novel characteristics of the system and environment modelled (see Section 5).

Section 2 provides an overview of MoSAIC and background information on erasure codes. Section 3 describes our methodology. Section 4 summarizes the results obtained and discusses their impact on the design of the cooperative backup service. Section 5 presents related work. Finally, Section 6 concludes on our findings and depicts future research directions.

## 2. Background

In this section, we present our work on the design of a cooperative backup service for mobile devices. Background information on erasure codes is then provided.

### 2.1. MoSAIC Overview

Our cooperative backup service, which we call MoSAIC, can leverage (i) excess storage resources available on mobile devices and (ii) short-range, high-bandwidth, and relatively energy-efficient wireless communications (Bluetooth, Zigbee, or Wi-Fi). More importantly, we expect our cooperative backup service to improve long-term availability of data produced by mobile devices. The idea is borrowed from peer-to-peer cooperative services: participating devices offer storage resources and doing so allows them to benefit from the resources provided by other devices in order to replicate their data [15]. Participating devices discover other devices in their vicinity using a suitable service discovery mechanism such as [9, 24] and communicate through single-hop connections, thereby limiting interactions to small physical regions.

Anyone is free to participate in the service and, therefore, participants have no prior trust relationship. In the sequel, we use the term *contributor* when referring to a node acting as a storage provider; we use the term *data owner* when referring to a "client" device, i.e., one that uses storage provided by the contributors to replicate its data. All participating devices may play *both* the owner and the contributor role.

When out of reach of Internet access and network infrastructure, devices meet and spontaneously form *ad hoc* networks which they can use to back-up data. However, it would be unrealistic to rely on chance encounters between devices for recovery. Instead, we require contributing devices to eventually send data stored on behalf of other devices to an Internet-based store accessible by the data owners [5, 15]. This way of handling intermittent infrastructure connectivity makes our approach comparable to delay-tolerant networks (DTNs): data blocks that are transmitted by data owners to contributors can be viewed as packets sent to the Internet-based store and where contributors act as relays [26, 31].

MoSAIC's approach to cooperative backup bears some similarity with earlier work on cooperative data storage [2, 12, 22] and caching for mobile devices [10, 21, 30]. However, it differs from them in several ways. First, unlike typical distributed file systems access patterns, data that is backed up is produced by a single device and may usually not be accessed by other devices. Second, unlike most caching strategies, our approach does not seek to improve locality of data replicas: in-

stead we expect replicas to propagate to the Internet-based store, much like packets in a DTN.

Previous work studied the design of a storage layer for our cooperative backup service [5] as well as that of security primitives enabling secure cooperation among mutually suspicious principals [3]. This paper focuses on the dependability evaluation of the proposed approach. The work presented here is two-fold: it aims to determine influential environmental parameters and their impact on data dependability, and compares different data scattering and replication strategies, including the use of erasure coding, in this cooperative backup context.

## 2.2. Erasure Codes

Erasure coding algorithms have been studied extensively [17, 20, 27, 28, 29]. Here we do not focus on the algorithms themselves but on their properties. A commonly accepted definition of erasure coding is the following [17, 28]:

- Given a  $k$ -symbol input datum, an erasure coding algorithm produces  $n \geq k$  fragments.
- $m$  fragments are necessary and sufficient to recover the original datum, where  $k \leq m \leq n$ . When  $m = k$ , the erasure code algorithm is said to be *optimal* [29].

Although not all erasure coding algorithms are optimal (many of them are *near-optimal* [29]), we will assume in the sequel the use of an optimal erasure code where  $m = k$ . By convention, we note  $(n,k)$  such an optimal erasure code [28].

When all  $k$  fragments are stored on different devices, an optimal erasure code allows  $n - k$  failures (or erasures) to be tolerated (beside that of the primary replica). The storage cost (or *stretch factor*) for an optimal erasure code is  $n/k$  (the inverse ratio  $k/n$  is often called the *rate* of an erasure code). To tolerate a number of erasures  $f$ , we need  $n = k + f$ , so the storage cost is  $1 + f/k$ . Therefore, erasure coding (with  $k \geq 2$ ) is more storage-efficient than simple replication ( $k = 1$ ). For instance, (2,1) and (3,2) erasure codes can both allow the tolerance of one failure, but the former requires twice as much storage as the original data while the latter only requires 1.5 times as much.

Additionally, when all  $k$  fragments are distributed to different devices belonging to different non-colluding users (or under different administrative domains), erasure codes can be regarded as a means for improving data confidentiality: to access the data, an attacker must have control over  $k$  contributing devices instead of just one when simple replication is used [8]. This effectively raises the bar for confidentiality attacks and may usefully complement ciphering techniques used at other layers. Similar concerns are addressed by generalized thresh-

old schemes where, in addition to the definition above, less than  $p \leq k$  fragments convey no information about the original data, from an information-theoretic viewpoint [11].

## 3. Methodology

In this section, we present the approach we have taken to model and evaluate our cooperative backup service. In particular, we describe and discuss the characteristics of the system modeled. We then present our use of Markov chains and the evaluated dependability measures.

### 3.1. System Characteristics

The cooperative backup service that we model is characterized by its replication and scattering strategy, and the considered device-to-device and device-to-Internet backup opportunities.

#### 3.1.1. Replication Strategy

Our model considers the case of a data owner that needs to replicate a single data item (generalization of the results to more than one data item is straightforward). We consider that the owner follows a *pre-defined* replication and dissemination strategy, using  $(n,k)$  erasure coding, where  $n$  is decided off-line, *a priori*, and where the owner distributes one and only one fragment to each encountered contributor. When  $k = 1$ , the strategy corresponds to simple replication. In practice, the exact choice of  $n$  and  $k$  could be made as a function of the scenario's characteristics and the user's dependability and confidentiality requirements.

This replication strategy privileges confidentiality over data dependability: only one fragment is given to each contributor encountered<sup>1</sup>, at the risk of being unable to distribute all the fragments in the end (for instance, because not enough contributors become available). An alternative strategy that favors data dependability over confidentiality consists in providing a contributor with as many fragments as possible while it is reachable.

Furthermore, the replication strategy is considered *static*. In particular, we consider that owners are not aware of the failures of contributors storing data on their behalf. Thus, owners cannot, for instance, decide to create more replicas when previously encountered contributors have failed. The fact that owners cannot be made aware of contributor failures is realistic under most of the scenarios envisaged: first, detection of the failure of a participating device would be hard to achieve in a mo-

---

<sup>1</sup> According to this policy, a contributor encountered more than once will only be given a fragment the first time it is encountered.

mobile context where nodes continuously come and go, and second, since devices are mobile, they are likely to be out of reach at the time the contributor fails.

### 3.1.2. Backup Opportunities

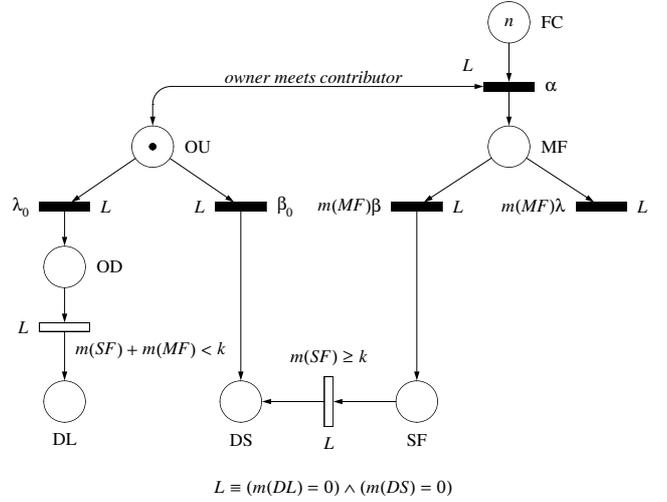
We consider that every encounter between devices offers a backup opportunity. Specifically, every device encountered is considered to be a contributor that *unconditionally accepts* storage requests from the data owner. Data owners unconditionally send one data fragment to each contributor encountered. Note that scenarios in which not all encounters offer backup opportunities (e.g., with contributors refusing to cooperate) can be simply modeled by introducing an opportunity/encounter ratio as an additional parameter.

We consider that Internet connection is only exploited when it is cheap and provides a high bandwidth. Thus, whenever a node gains Internet access, we assume that it transfers all the data fragments it currently stores on behalf of other nodes.

### 3.2. Modeling Approach

Two complementary techniques can be used to support model-based evaluation approaches: analytical techniques and simulation. Analytical techniques are commonly used to support dependability evaluation studies. They allow one to obtain mathematical expressions of the relevant measures, which can then be explored to easily identify trends and to carry out sensitivity analysis. When using analytical techniques, the system must be described at a high level of abstraction. Simplifying assumptions are generally needed to obtain tractable models. Although simulation can be used to describe the system at a more detailed level, it is more costly in terms of the processing time needed to obtain accurate and statistically significant quantitative results.

Markov chains and generalized stochastic Petri nets (GSPNs) are commonly used to perform dependability evaluation studies and sensitivity analyses aimed at identifying parameters having the most significant impact on the measures. The corresponding models are based on the assumption that all the underlying stochastic processes are described by exponential distributions. Although such an assumption may not faithfully reflect reality, the results obtained from the models and sensitivity analysis give preliminary indications and estimations about the expected behaviors and trends that can be observed. More accurate results can be obtained considering more general distributions, using for example the “stages method” [6] or non Markovian models. However, in this paper, we assume that all stochastic processes are exponentially distributed.



**Figure 1. Petri net of the replication and scattering process for an  $(n,k)$  erasure code.**

In the following, we present a generic GSPN model and the corresponding Markov chains. Then, we present the quantitative measures evaluated from the models to assess data dependability. Finally, we discuss the main parameters that are considered in the sensitivity analysis studies.

### 3.3. GSPN and Markov Models

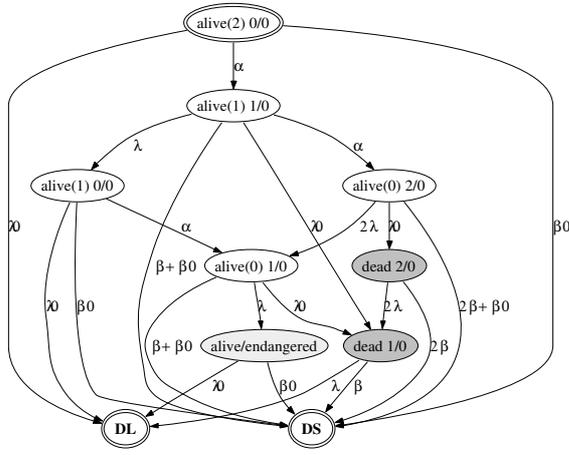
GSPNs are widely used to support the construction of performance and dependability evaluation models based on Markov chains. In particular, they are well suited to represent synchronization, concurrency and dependencies among processes [18].

Figure 1 presents the GSPN model of MoSAIC using an  $(n,k)$  erasure coding algorithm.

The model focuses on the mobile *ad hoc* part of the cooperative backup service, purposefully ignoring issues related to the implementation of the Internet-side functionalities. Thus, a data fragment is considered “safe” (i.e., it cannot be lost) whenever either its owner or a contributor storing it is able to access the Internet. In other words, the Internet-based store of our cooperative backup service is abstracted as a “reliable store”. Conversely, if a participating device fails before reaching the Internet, then all the fragments it holds are considered lost.

Thus, with  $(n,k)$  erasure coding, a data item is definitely lost if and only if its owner device fails *and* less than  $k$  contributors hold or have held a fragment of the data item.

Our model consists of three main processes represented by timed transitions with constant rate exponential distributions:



**Figure 2. Markov chain of the replication and scattering process for a (2,1) erasure code.**

- A process with rate  $\alpha$  that models the encounter of a contributor by the data owner, where the owner sends one data fragment to the contributor.
- A process that models the connection of a device to the Internet, with rate  $\beta_0$  for the owner and  $\beta$  for contributors.
- A process that represents the failure of a device, with rate  $\lambda_0$  for the owner and  $\lambda$  for contributors.

The GSPN in Figure 1 is divided into two interacting subnets. The subnet on the left describes the evolution of a data item at the owner device: either it is lost (with rate  $\lambda_0$ ), or it reaches the on-line reliable store (with rate  $\beta_0$ ). Places OU and OD denote situations where the owner device is “up” or “down”, respectively. The subnet on the right describes: (i) the data replication process leading to the creation of “mobile fragments” (place MF) on contributor devices as they are encountered (with rate  $\alpha$ ), and (ii) the processes leading to the storage of the fragments (place SF) in the reliable store (rate  $\beta$ ), or its loss caused by the failure of the contributor device (rate  $\lambda$ ). At the top of the right-hand side subnet is place FC whose initial marking denotes the number of fragments to create. The transition rates associated with the loss of a data fragment or its storage on the Internet are weighted by the marking of place MF, i.e., the number of fragments that can enable the corresponding transitions.

Two places with associated immediate transitions are used in the GSPN to identify when the data item is safely stored in the reliable store (place DS), or is definitely lost (place DL), respectively. The “data safe” state is reached (i.e., DS is marked) when the original data item from the owner node or at least  $k$  fragments from

the contributors reach the Internet. The “data loss” state is reached (i.e., DL is marked) when the data item from the owner node is lost and less than  $k$  fragments are available. This condition is represented by a predicate associated with the immediate transition that leads to DL. Finally,  $L$  is the GSPN “liveliness predicate”, true if and only if  $m(DS) = m(DL) = 0$ : as soon as either DS or DL contains a token, no transition can be fired.

The GSPN model of Figure 1 is generic and can be used to automatically generate the Markov chain associated with any  $(n,k)$  erasure code. As an example, Figure 2 shows the Markov chain representing replication and scattering with a (2,1) erasure code (that is, simple replication). The arrow labels represent the rate of the corresponding transition; for instance, an arrow labeled “ $2\lambda$ ” represents the failure of one of the two contributors holding a copy of the owner’s data. On this graph, we distinguish four sets of states:

1. The states labeled “alive( $X$ )  $Y/Z$ ” denote those where the owner device is available, where  $X$ ,  $Y$  and  $Z$  represent, respectively, the number of fragments left to distribute, the number of fragments held by contributors, and the number of fragments available in the on-line reliable store (i.e., “safe”). Here,  $k \leq X + Y + Z \leq n$  and  $Z < k$  (for the (2,1) erasure code presented on Figure 2,  $Z$  is always zero).
2. States labeled “dead  $Y/Z$ ” represent those where the owner device has failed. Here,  $Y + Z \geq k$  and  $Z < k$ .
3. The state labeled “alive/endangered” aggregates all states where the owner device is available and where  $X + Y + Z < k$ . In this situation, although fragments may be available in the reliable store and on contributing devices, failure of the owner device yields definite data loss.
4. The two absorbing states, labeled “DS” (“data safe”) and “DL” (“data lost”), represent, respectively, the safe arrival of the data item in the reliable store and its definite loss.

Similar loop-free Markov chains can be generated for any  $(n,k)$ . These families of states were devised as a result of our analysis of the system modeled. The total number of states in such an  $(n,k)$  Markov chain is  $O(n^2)$ . The models we are considering, with reasonably small values of  $n$  (i.e., small storage overhead), are tractable using available modeling tools.

### 3.4. Quantitative Measures

We analyze the dependability of our backup service via the probability of data loss, i.e., the asymptotic probability, noted  $PL$ , of reaching the “data lost” state. For a given erasure code  $(n,k)$ ,  $PL$  can be easily evaluated from

the corresponding Markov chain using well-known techniques for absorbing Markov chains [13]. The smaller  $PL$  is, the more dependable is the data backup service.

To measure the dependability improvement offered by MoSAIC, we compare  $PL$  with the probability of data loss  $PL_{ref}$  of a comparable, non-MoSAIC scenario where:

- data owners do not cooperate with other mobile devices;
- data owner devices fail with rate  $\lambda_0$ ;
- data owners gain Internet access and send their data items to a reliable store with rate  $\beta_0$ .

This scenario is modeled by a simple Markov chain where the owner's device can either fail and lose the data or reach the Internet and save the data. The probability of loss in this scenario is:  $PL_{ref} = \frac{\lambda_0}{\lambda_0 + \beta_0}$ .

We note  $LRF$  the data loss probability reduction factor offered by MoSAIC compared to the above non-MoSAIC scenario, where  $LRF = PL_{ref}/PL$ . The higher  $LRF$ , the more MoSAIC improves data dependability. For instance,  $LRF = 100$  means that data on a mobile device is 100 times more unlikely to be lost when using MoSAIC than when not using it.

### 3.5. Parameters

$PL$  and  $LRF$  depend on a number of parameters ( $n$ ,  $k$ ,  $\alpha$ ,  $\beta$ ,  $\lambda$ ,  $\beta_0$ , and  $\lambda_0$ ). Rather than considering absolute values for the rates of stochastic processes, we consider ratios of rates corresponding to pertinent competing processes.

For example, the usefulness of cooperative backup will depend on the rates at which contributing devices are met relative to the rate at which connection to the fixed infrastructure is possible. Therefore, it makes sense to study  $LRF$  as a function of the *ad hoc*-to-Internet connectivity ratios  $\frac{\alpha}{\beta}$  and  $\frac{\alpha}{\beta_0}$  rather than for absolute values of these parameters.

The ratios  $\frac{\beta}{\lambda}$  and  $\frac{\beta_0}{\lambda_0}$  can be thought of as the *effectiveness* of the contributors and data owner towards data backup. A device that fails frequently but which often connects to Internet could be as effective as one that fails rarely, but also connects infrequently. We will study the evolution of dependability improvement when the effectiveness of contributors varies relative to that of the data owner. This allows us to study the impact of contributor ineffectiveness (leading to  $\lambda > \lambda_0$ ). For most measures, however, we will assume that  $\beta = \beta_0$  and  $\lambda = \lambda_0$ .

Finally, one may question the assumption that contributors accept *all* requests, at rate  $\alpha$ , regardless of their amount of available resources. However, simple back-of-the-envelope calculations provide evidence that this

is a reasonable assumption. When the replication strategy described in Section 3.1.1 is used, the number of fragments (i.e., storage requests) that a contributor may receive during the time between two consecutive Internet connections is, on average,  $\frac{\alpha}{\beta}$ . Let  $s$  be the size of a fragment: a contributor needs, on average,  $V = s \left(\frac{\alpha}{\beta}\right)$  storage units to serve all these requests. If a contributor's storage capacity,  $C$ , is greater than  $V$ , it can effectively accept all requests; otherwise, the contributor is *saturated* and can no longer accept any storage request.

In other words, redefining  $\alpha$  as the *effective* encounter rate (i.e., the rate of encounters of contributors that accept storage requests), and letting  $\gamma$  be the *actual* encounter rate, we have:  $\frac{\alpha}{\beta} = \min\left(\frac{\gamma}{\beta}, \frac{C}{s}\right)$ . A realistic estimate with  $C = 2^{30}$  (contributor storage capacity of 1 GB) and  $s = 2^{10}$  (fragment size of 1 KB) shows that contributors would only start rejecting requests when  $\frac{\gamma}{\beta} > 2^{20}$ , a ratio that is beyond most realistic scenarios.

## 4. Results

This section presents and discusses the results of our analysis.

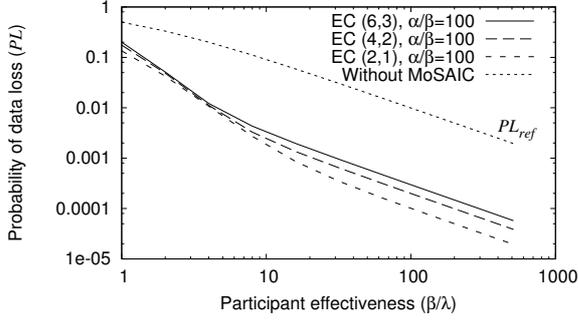
### 4.1. Overview

Figure 3 shows the absolute probability of data loss without MoSAIC ( $PL_{ref}$ , as defined earlier), and with MoSAIC, using various erasure code parameters. In the MoSAIC case, the *ad hoc*-to-Internet connectivity ratio  $\frac{\alpha}{\beta}$  is fixed at 100. Both with and without MoSAIC, the probability of loss falls to zero as  $\frac{\beta}{\lambda}$  tends to infinity; however, using MoSAIC, the probability of loss falls below 0.01 with  $\frac{\beta}{\lambda} \approx 6$ , while it requires  $\frac{\beta}{\lambda} \approx 100$  to reach the same value without MoSAIC.

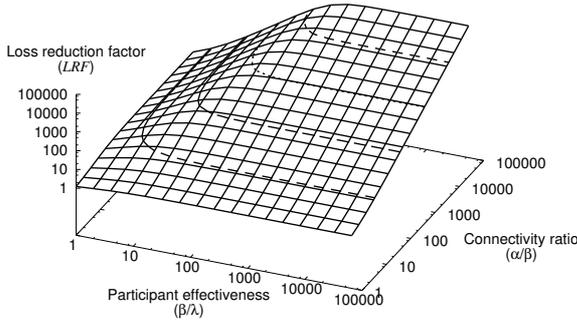
Figure 4 shows the data dependability improvement yielded by MoSAIC with a (2,1) erasure code using the replication strategy outlined in Section 3.1.1. Here, we assume that contributors and owners behave identically, i.e.,  $\beta_0 = \beta$  and  $\lambda_0 = \lambda$ .

Three observations can be made from this plot. First, as expected, the cooperative backup approach is not very relevant compared to the reference backup approach when  $\frac{\alpha}{\beta} = 1$  (i.e., when Internet access is as frequent as *ad hoc* encounters). Figure 5 shows the contour lines of  $LRF$  extracted from Figure 4: for the cooperative backup approach to offer at least an order of magnitude improvement over the reference backup scheme, the environment must satisfy  $\frac{\beta}{\lambda} > 2$  and  $\frac{\alpha}{\beta} > 10$ .

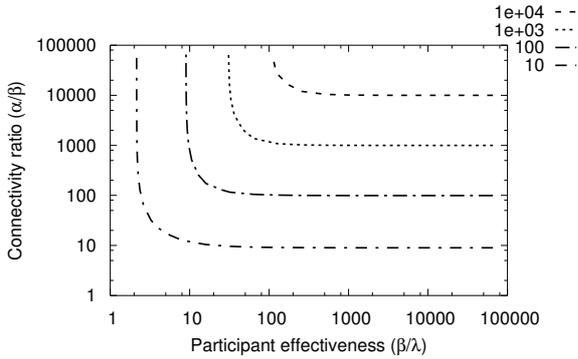
Second, for any given  $\frac{\alpha}{\beta}$ ,  $LRF$  reaches an asymptote after a certain  $\frac{\beta}{\lambda}$  threshold. Thus, for any given connectivity ratio  $\frac{\alpha}{\beta}$ , increasing the infrastructure connectivity to failure rate ratio  $\frac{\beta}{\lambda}$  is only beneficial up to that threshold.



**Figure 3. Data loss probability with and without MoSAIC.**



**Figure 4. Loss reduction factor  $LRF$  for a (2,1) erasure code.**



**Figure 5. Contour map of  $LRF$  for Figure 4.**

The third observation that can be made is that the dependability improvement factor first increases propor-

tionally to  $\alpha/\beta$ , and then, at a certain threshold, rounds off towards an asymptote (visible on Figure 4 for small values of  $\beta/\lambda$  but hidden for high values due to choice of scale). Other  $(n,k)$  plots have a similar shape.

## 4.2. Asymptotic Behavior

Figure 6 shows  $LRF$  as a function of  $\beta/\lambda$ , for different values of  $\alpha/\beta$  and different erasure codes (again, assuming the data owner's failure and connection rates are the same as those of contributors). This again shows that the maximum value of  $LRF$  for any erasure code, as  $\beta/\lambda$  tends to infinity, is a function of  $\alpha/\beta$ . We verified the following formula for a series of codes with  $n \in \{2, 3, 4, 5\}$  and  $k \in \{1, 2, 3\}$  (with  $k \leq n$ ) and postulate that it is true for all positive values of  $n$  and  $k$  such that  $n \geq k$ :

$$(4.0)$$

This equation can be simplified as follows:

$$(4.0)$$

First, it describes an asymptotic behavior, which confirms our initial numerical observation. Second, it does not depend on  $n$ . This observation provides useful insight in how to choose the most appropriate erasure coding parameters, as we will see in Section 4.3.

We can similarly compute the limiting value of  $LRF(n, k)$  as  $\alpha/\beta$  tends to infinity:

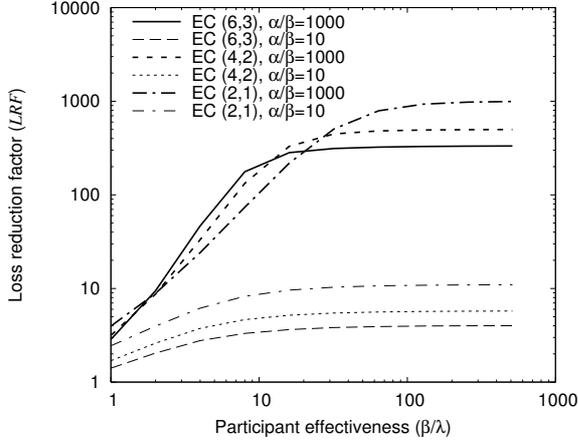
$$\lim_{\beta/\lambda \rightarrow \infty} \left( LRF_{n,k} \left( \frac{\alpha}{\beta}, \frac{\beta}{\lambda} \right) \right) = \frac{\left( 1 + \frac{\alpha}{\beta} \right)^k}{\sum_{x=0}^{k-1} \binom{k}{x} \left( \frac{\alpha}{\beta} \right)^x}$$

$$\lim_{\alpha/\beta \rightarrow \infty} \left( LRF_{n,k} \left( \frac{\alpha}{\beta}, \frac{\beta}{\lambda} \right) \right) = \frac{\left( 1 + \frac{\beta}{\lambda} \right)^n}{\sum_{x=0}^{k-1} \binom{n}{x} \left( \frac{\beta}{\lambda} \right)^x} \quad (4.3)$$

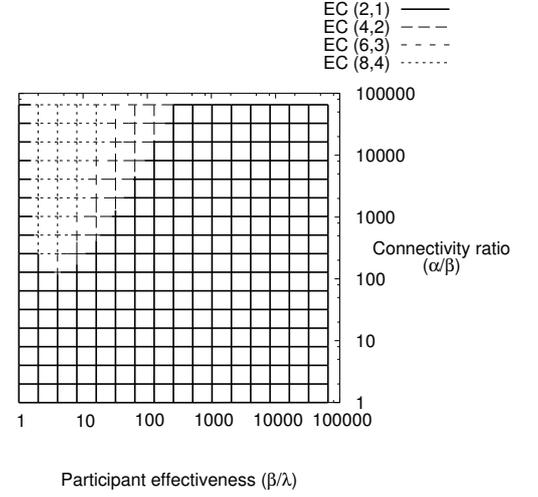
This shows that  $LRF$  also reaches an asymptote as  $\alpha/\beta$  grows, and that the value of this asymptote is dependent on  $\beta/\lambda$ .

## 4.3. Erasure Coding vs. Simple Replication

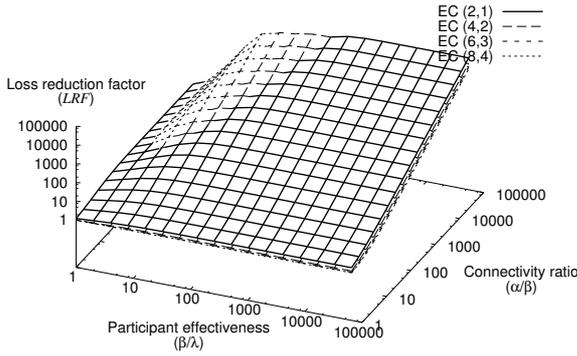
Figure 6 allows us to compare the improvement factor yielded by MoSAIC as different erasure codes are used. The erasure codes shown on the plot all incur the same storage cost:  $\frac{n}{k} = 2$ . In all cases, the maximum dependability improvement decreases as  $k$  increases. This is confirmed analytically by computing the following ratio, for any  $p > 1$  such that  $pk$  and  $pn$  are integers:



**Figure 6. Loss reduction factor for different erasure codes.**



**Figure 8. Comparing  $LRF$  for different erasure codes with  $\frac{n}{k} = 2$ : projection.**



**Figure 7. Comparing  $LRF$  for different erasure codes with  $\frac{n}{k} = 2$ .**

$$R_p = \frac{\lim_{\frac{\beta}{\lambda} \rightarrow \infty} \left( LRF_{pn,pk} \left( \frac{\alpha}{\beta}, \frac{\beta}{\lambda} \right) \right)}{\lim_{\frac{\beta}{\lambda} \rightarrow \infty} \left( LRF_{n,k} \left( \frac{\alpha}{\beta}, \frac{\beta}{\lambda} \right) \right)} = \frac{1 - \left( \frac{\alpha}{\beta} \right)^k}{1 - \left( \frac{\alpha}{\beta} \right)^{kp}} \quad (4.4)$$

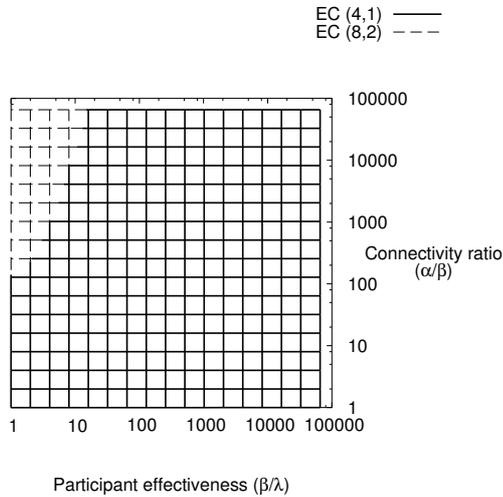
We see that  $R_p < 1$  for  $p > 1$ . Thus, we conclude that, from the dependability viewpoint, simple replication (i.e., with  $k = 1$ ) is *always* preferable to erasure coding (i.e., with  $k > 1$ ) above a certain  $\frac{\beta}{\lambda}$  threshold.

Different trends can be observed for lower values of  $\frac{\beta}{\lambda}$ . As illustrated on Figures 6 and 7, we can numerically compare the dependability yielded by various era-

sure codes. Figure 7 compares the dependability improvement yielded by several erasure codes having the same storage cost; only the top-most erasure code (i.e., the surface with the highest  $LRF$ ) is visible from above. The (2,1) plot is above all other plots, except in a small region where the other erasure codes (thin dashed and dotted lines) yield a higher  $LRF$ .

Figure 8, which is a projection of this 3D plot on the  $\frac{\beta}{\lambda}$  and  $\frac{\alpha}{\beta}$  plane, allows the visualization of the region where erasure codes perform better than simple replication. The solid lines covering a large surface show that simple replication (a (2,1) code) yields better dependability than erasure coding in a large spectrum of scenarios. Erasure codes yield a higher data dependability than simple replication in the region defined (roughly) by  $\frac{\alpha}{\beta} > 100$  and  $1 < \frac{\beta}{\lambda} < 100$ . However, in this region, the dependability yielded by erasure codes is typically less than an order of magnitude higher than that yielded by simple replication, even for the (extreme) case where  $\frac{\alpha}{\beta} = 1000$  (see Figure 6).

Interestingly, similar plots obtained for larger values of  $\frac{n}{k}$  (e.g., see Figure 9) show that the region where erasure codes prevail tends to shift towards lower  $\frac{\beta}{\lambda}$  values as  $\frac{n}{k}$  increases. In other words, the spectrum of scenarios where erasure codes provide better dependability than simple replication narrows as the chosen storage overhead (the  $\frac{n}{k}$  ratio) increases.



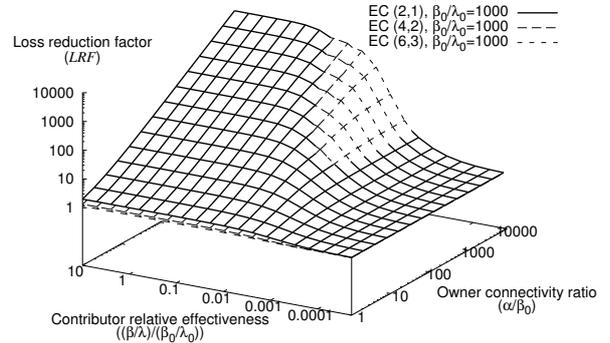
**Figure 9. Comparing  $LRF$  for different erasure codes with  $\frac{n}{k} = 4$ : projection.**

Nevertheless, when confidentiality is an important criterion, using erasure coding instead of simple replication is relevant. Erasure coding can achieve better confidentiality than simple replication [8] at the cost of a slightly lower asymptotic dependability improvement factor. For instance, in the context of Figure 6, if the user wants to maximize confidentiality while requiring a minimum improvement factor of 100, a (6,3) erasure code would be chosen rather than simple replication.

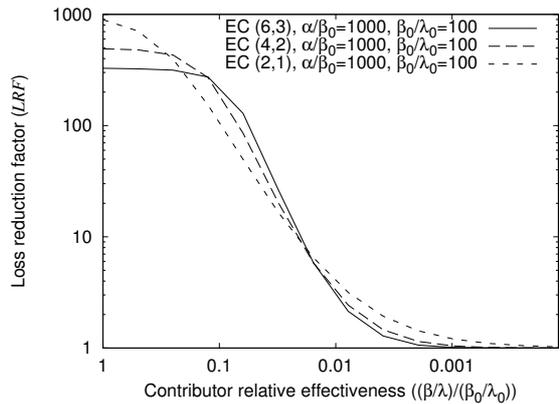
#### 4.4. Impact of Contributor Effectiveness

The fact that contributors may be less reliable than the data owner can be studied by distinguishing the owner's Internet access and failure rates ( $\beta_0$  and  $\lambda_0$ ) from the contributor's ( $\beta$  and  $\lambda$ ). Since the cooperative backup service is open to anyone willing to participate, participants do not trust each other *a priori*. It may be the case that some contributors are malicious or otherwise largely unreliable, thereby proving to be less *effective*, with respect to accessing the on-line reliable store, than the data owner, as mentioned in Section 3.5. We define the *effectiveness* of a contributor (respectively, the data owner) as a backup agent by the ratio  $\frac{\beta}{\lambda}$  (respectively  $\frac{\beta_0}{\lambda_0}$ ). Thus, the effectiveness of a contributor relative to that of the data owner can be expressed by  $\frac{\beta/\lambda}{\beta_0/\lambda_0}$ .

Figure 10 shows  $LRF$  as a function of the contributor-to-owner effectiveness ratio on one hand,



**Figure 10. Loss reduction factor as contributors become less effective.**



**Figure 11. Data availability improvement as contributors become less effective for a given  $\frac{\alpha}{\beta_0}$ .**

and the owner's *ad hoc* to Internet connectivity ratio on the other. At the left end of the spectrum, where contributors are as effective or even more so than the owner itself, MoSAIC is very beneficial in terms of backup dependability. In this scenario, the data loss reduction improvement is proportional to  $\alpha_{\beta_0}$  up to an asymptote, as was already discussed in the previous sections.

Not surprisingly, at the other end of the spectrum, the curve shows that as contributors become less effective than the data owner,  $LRF$  decreases until it becomes equivalent to the non-MoSAIC case (i.e.,  $LRF = 1$ ). More precisely, when contributors become 100 times less effective than the owner, the loss reduction improvement factor yielded by MoSAIC is less than 10. The

threshold at which this occurs does not depend on the value of  $\alpha_{\beta_0}$ .

Furthermore, Figure 10 shows that the threshold is the same for all erasure codes. However, again, there is a small region of the plot where erasure codes are more resilient against contributor ineffectiveness than simple replication. For this setup, the region where erasure codes prevail over simple replication is limited to effectiveness ratios between 0.1 and 0.001 and  $\alpha_{\beta_0}$  above 100. With  $\frac{\alpha}{\beta} = 1000$ , the gain provided by erasure codes in this region is less than an order of magnitude, as shown on Figure 11. Thus, for many use cases of MoSAIC, using erasure coding instead of simple replication will not provide appreciable improvement in the face of misbehaving contributors, from the dependability point of view.

## 5. Related Work

There is a large literature dealing with the design of peer-to-peer file sharing architectures and cooperative backup services in mobile and fixed infrastructures. A recent survey can be found in [14]. In this section we focus on related work dealing with the dependability evaluation of replication mechanisms, notably erasure coding algorithms.

Erasure coding has been considered in the framework of delay-tolerant networks. Evaluations of their impact on message delivery time have been carried through simulation [26]. However, to our knowledge, no analytical evaluation of the probability of message delivery has been made.

Several papers analyze data dependability in distributed and peer-to-peer storage systems. The authors of OceanStore conducted an analytical evaluation of the MTTF (mean time to failure) of a distributed, self-repairing storage system [27]. They conclude that erasure codes yield MTTF orders of magnitude higher than simple replication; however, their computations are based on probability distributions of hard disk failures, which may be quite different from that of individual untrusted peers on the Internet. A similar comparison for peer-to-peer storage is proposed in [25], using a stochastic model. They conclude on the unsuitability of erasure codes in a peer-to-peer environment where peer availability is low. The major difference between these studies and what we present here is that the authors model a data block *repair* process that is inexistent in the context of a mostly-disconnected peer-to-peer backup system, notably because data owners cannot be made aware of contributor failures.

In [17], the authors analyze erasure code replication and compare the resulting data availability as a function of individual host availability (assuming each host stores

exactly one fragment of the original data) and erasure code parameters  $(n,k)$ . They identify a “switch point” between scenarios where erasure coding is preferable (from a data availability viewpoint) and scenarios where simple replication should be used. More precisely, they conclude that simple replication yields better data availability when host availability is low.

Our results comparing erasure codes and simple replication in terms of dependability are in agreement with those obtained on simpler models [1, 17, 25]. We observe a switch point similar to that of [17]. For instance, in our model, whether erasure codes yield better data dependability than simple replication depends on  $\alpha_{\beta}$  and  $\beta_{\lambda}$  (see, e.g., Figure 7).

Building on a similar analysis, *TotalRecall* [1], a peer-to-peer storage system, proposes mechanisms to automate availability management, which includes dynamic parameterization of erasure coding replication based on predicted host availability. However, the authors do not consider the use of erasure codes as a means to improve data confidentiality [8]. Additionally, the mobile environment we are addressing leads to a wider range of scenarios (and connectivity). A dynamic replication strategy for peer-to-peer cooperative data storage among untrusted nodes is also presented in [23], though they do not consider the use of erasure codes.

Finally, our partners within the MoSAIC project at IRISA have taken an approach that is complementary to ours. They provide formulæ that may be used for on-line assessment of the impact of the distribution of a single fragment on the overall data availability [19]. Assuming the probability of successfully restoring individual fragments can be estimated, these formulæ may be used as a hint by the backup software to improve replica scheduling.

## 6. Conclusion and Future Work

Our evaluation based on Petri nets and Markov chains allowed us to achieve our goals (see Section 1). First, we have a better understanding of the scenarios where MoSAIC yields noticeable data dependability improvement. Namely, we showed that the cooperative backup approach is beneficial (i.e., yields data dependability an order of magnitude higher than without MoSAIC) only when  $\frac{\beta}{\lambda} > 2$  and  $\frac{\alpha}{\beta} > 10$ . We demonstrated that MoSAIC can decrease the probability of data loss by a factor that can be as large as the *ad hoc* to Internet connectivity ratio  $\alpha_{\beta}$ . We also showed that the cooperative backup approach does not improve data dependability when contributors are more than 100 times less effective than data owners. This result will have to be taken into account in the design of cooperation incentive and trust mechanisms [3, 15].

Second, we compared simple replication and erasure codes and concluded that erasure codes provide an advantage (dependability-wise) over simple replication only in narrow scenarios. Those scenarios are restricted to low  $\beta_\lambda$  and high  $\alpha_\beta$  and the dependability improvement provided by erasure codes in those cases is typically less than an order of magnitude. Measurements of actual use cases are needed in order to see what real-world situations these scenarios map to.

Based on our results, several replication strategies can be envisioned. One possible strategy would be to maximize data dependability for a given user-specified storage overhead. Since in most scenarios little can be gained from using erasure codes, and since the consequence of a wrong decision would be detrimental to data dependability (e.g., choosing erasure coding in a scenario where simple replication would have been more beneficial), the best way to maximize data dependability is to always use simple replication.

Other replication strategies can be imagined. Instead of focusing only on dependability, users may specify additional fragmentation to increase confidentiality [8]. Such a strategy could maximize fragmentation (i.e., by choosing a high  $k$  value) according to environmental parameters, while honoring a user-specified minimum dependability improvement factor. The *ad hoc* and Internet connectivity rates could be estimated, for instance, by collecting actual data about single-hop device encounters of a device that is carried around according to some mobility scenario. These environmental parameters as well as the effectiveness of encountered contributors could also be estimated on-line based on past observations, perhaps augmented by user input, and used as a hint to the replication strategy.

Current and future work also includes refining our model to allow for the distribution of more than one fragment per contributor. Doing so would allow the evaluation of a wider range of replication and dissemination strategies. So-called *rate-less* erasure codes allow the production of an unlimited number of distinct fragments, out of which any  $k$  suffice to recover the original data [20]. Their use could also be evaluated with little impact on our model, for instance by choosing higher values of parameter  $n$ .

The evaluation work presented in this paper is part of a larger design and implementation effort of a cooperative backup service for mobile devices. Our earlier work explored other aspects of the design space, particularly relating to storage tradeoffs and data encoding and compression techniques [5]. Other issues are to be addressed in this framework, notably with respect to the integration of *ad hoc* networking techniques and trust management [3, 15].

## Acknowledgements

We thank our colleagues Thomas Robert and Benjamin Lussier for their insightful and constructive comments on an earlier version of this paper.

## References

- [1] R. BHAGWAN, K. TATI, Y-C. CHENG, S. SAVAGE, G. M. VOELKER. Total Recall: System Support for Automated Availability Management. In *Proc. of the ACM/USENIX NSDI*, March 2004.
- [2] M. BOULKENAFED, V. ISSARNY. AdHocFS: Sharing Files in WLANs. In *Proc. of the 2nd Int. Symp. on Network Computing and Applications*, April 2003.
- [3] L. COURTÈS, M-O. KILLIJIAN, D. POWELL. Security Rationale for a Cooperative Backup Service for Mobile Devices. In *Proc. of the Latin-American Symp. on Dependable Computing*, Springer-Verlag, 2007.
- [4] L. COURTÈS, M-O. KILLIJIAN, D. POWELL, M. ROY. Sauvegarde coopérative entre pairs pour dispositifs mobiles. In *Actes des deuxièmes journées francophones Mobilité et Ubiquité (UbiMob)*, pp. 97–104, ACM Press, May 2005.
- [5] L. COURTÈS, M-O. KILLIJIAN, D. POWELL. Storage Tradeoffs in a Collaborative Backup Service for Mobile Devices. In *Proc. of the 6th European Dependable Computing Conf.*, pp. 129–138, IEEE CS Press, October 2006.
- [6] D.R COX, H.D. MILLER. The Theory of Stochastic Processes. Chapman and Hall Ltd., 1965.
- [7] L. P. COX, C. D. MURRAY, B. D. NOBLE. Pastiche: Making Backup Cheap and Easy. In *5th USENIX OSDI*, pp. 285–298, December 2002.
- [8] Y. DESWARTE, L. BLAIN, J-C. FABRE. Intrusion Tolerance in Distributed Computing Systems. In *Proc. of the IEEE Symp. on Research in Security and Privacy*, pp. 110–121, May 1991.
- [9] P. ENGELSTAD, Y. ZHENG, T. JØNVIK, D. V. THANH. Service Discovery and Name Resolution Architectures for On-Demand MANETs. In *Proc. of the Int. Conf. on Distributed Computing Systems Workshops*, IEEE CS Press, May 2003.
- [10] J. FLINN, S. SINNAMOHIDEEN, N. TOLIA, M. SATYANARAYANAN. Data Staging on Untrusted Surrogates. In *Proc. of the USENIX FAST*, March 2003.
- [11] G. R. GANGER, P. K. KHOSLA, M. BAKKALOGLU, M. W. BIGRIGG, G. R. GOODSON, S. OGUZ, V. PANDURANGAN, C. A. N. SOULES, J. D. STRUNK, J. J. WYLIE. Survivable Storage Systems. In *Proc. of the DARPA Information Survivability Conf. & Exposition (DISCEX)*, pp. 184–195, IEEE CS Press, June 2001.
- [12] A. KARYPIDIS, S. LALIS. OmniStore: A System for Ubiquitous Personal Storage Management. In *Proc. of the Annual IEEE Int. Conf. on Pervasive Computing and Communications (PerCom)*, pp. 136–147, IEEE CS Press, March 2006.
- [13] J. G. KEMENY, J. L. SNELL. Finite Markov Chains. D. Van Nostrand Co., Inc., Princeton, New Jersey, USA, 1960.

- [14] M-O. KILLIJIAN, L. COURTÈS, D. POWELL. A Survey of Cooperative Backup Mechanisms. Technical Report LAAS Report 06472, LAAS-CNRS, October 2006.
- [15] M-O. KILLIJIAN, D. POWELL, M. BANÂTRE, P. COUDERC, Y. ROUDIER. Collaborative Backup for Dependable Mobile Applications. In *Proc. of 2nd Int. Workshop on Middleware for Pervasive and Ad-Hoc Computing (Middleware 2004)*, pp. 146–149, ACM Press, October 2004.
- [16] J. KUBIATOWICZ, D. BINDEL, Y. CHEN, S. CZERWINSKI, P. EATON, D. GEELS, R. GUMMADI, S. RHEA, H. WEATHERSPOON, W. WEIMER, C. WELLS, B. ZHAO. OceanStore: An Architecture for Global-Scale Persistent Storage. In *Proc. of the 9th ASPLOS*, pp. 190–201, November 2000.
- [17] W. K. LIN, D. M. CHIU, Y. B. LEE. Erasure Code Replication Revisited. In *Proc. of the 4th P2P*, pp. 90–97, 2004.
- [18] M.A. MARSAN, G. BALBO, G. CONTE, S. DONATELLI, G. FRANCESCHINIS. Modeling with Generalized Stochastic Petri Nets. John Wiley & Sons Ltd., 1995.
- [19] D. MARTIN-GUILLEREZ. Increasing Data Resilience of Mobile Devices with a Collaborative Backup Service. In *Supplemental Proc. of the Int. Conf. on Dependable Systems and Networks (DSN'06)*, pp. 139–141, 2006.
- [20] M. MITZENMACHER. Digital Fountains: A Survey and Look Forward. In *Proc. of the IEEE Information Theory Workshop*, pp. 271–276, October 2004.
- [21] M. PAPADOPOULI, H. SCHULZTRINNE. Seven Degrees of Separation in Mobile Ad Hoc Networks. In *IEEE Conf. on Global Communications (GLOBECOM)*, November 2000.
- [22] N. PREGUICA, C. BAQUERO, J. L. MARTINS, M. SHAPIRO, P. S. ALMEIDA, H. DOMINGOS, V. FONTE, S. DUARTE. FEW: File Management for Portable Devices. In *Proc. of the Int. Workshop on Software Support for Portable Storage*, March 2005.
- [23] K. RANGANATHAN, A. IAMNITCHI, I. FOSTER. Improving Data Availability Through Dynamic Model-Driven Replication in Large Peer-to-Peer Communities. In *Proc. of the Workshop on Global and Peer-to-Peer Computing on Large Scale Distributed Systems*, pp. 376–381, IEEE CS Press, May 2002.
- [24] F. SAILHAN, V. ISSARNY. Scalable Service Discovery for MANET. In *Proc. of the IEEE Int. Conf. on Pervasive Computing and Communication*, March 2005.
- [25] A. VERNOS, G. UTARD. Data Durability in Peer to Peer Storage Systems. In *Proc. of the 4th Workshop on Global and Peer to Peer Computing*, pp. 90–97, IEEE CS Press, April 2004.
- [26] Y. WANG, S. JAIN, M. MARTONOSI, K. FALL. Erasure-Coding Based Routing for Opportunistic Networks. In *Proc. of the ACM SIGCOMM Workshop on Delay-Tolerant Networking*, pp. 229–236, ACM Press, 2005.
- [27] H. WEATHERSPOON, J. KUBIATOWICZ. Erasure Coding vs. Replication: A Quantitative Comparison. In *Revised Papers from the 1st Int. Workshop on Peer-to-Peer Systems*, pp. 328–338, Springer-Verlag, 2002.
- [28] L. XU. Hydra: A Platform for Survivable and Secure Data Storage Systems. In *Proc. of the ACM Workshop on Storage Security and Survivability*, pp. 108–114, ACM Press, November 2005.
- [29] L. XU, V. BOHOSSIAN, J. BRUCK, D. G. WAGNER. Low Density MDS Codes and Factors of Complete Graphs. In *IEEE Transactions on Information Theory*, 45(1), November 1999, pp. 1817–1826.
- [30] L. YIN, G. CAO. Supporting Cooperative Caching in Ad Hoc Networks. In *IEEE Transactions on Mobile Computing*, 5(1), January 2006, pp. 77–89.
- [31] Z. ZHANG. Routing in Intermittently Connected Mobile Ad Hoc Networks and Delay Tolerant Networks: Overview and Challenges. In *IEEE Communications Surveys & Tutorials*, 8, January 2006, pp. 24–37.