Hybrid Approaches for Distributed Storage Systems

Julio Araujo, Frédéric Giroire, Julian Monteiro

Hybrid Approaches for Distributed Storage Systems*

Julio Araujo[1,2], Frédéric Giroire[1], and Julian Monteiro[3]

[1] Mascotte, joint project I3S(CNRS/UNS)/INRIA, France
[2] ParGO Research Group, Federal University of Ceará, Brazil
[3] Department of Computer Science, IME, University of São Paulo, Brazil

Abstract. Distributed or peer-to-peer storage solutions rely on the introduction of redundant data to be fault-tolerant and to achieve high reliability. One way to introduce redundancy is by simple replication. This strategy allows an easy and fast access to data, and a good bandwidth efficiency to repair the missing redundancy when a peer leaves or fails in high churn systems.
However, it is known that erasure codes, like Reed-Solomon, are an efficient solution in terms of storage space to obtain high durability when compared to replication.
Recently, the Regenerating Codes were proposed as an improvement of erasure codes to better use the available bandwidth when reconstructing the missing information.
In this work, we compare these codes with two hybrid approaches. The first was already proposed and mixes erasure codes and replication. The second one is a new proposal that we call Double Coding. We compare these approaches with the traditional Reed-Solomon code and also Regenerating Codes from the point of view of availability, durability and storage space. This comparison uses Markov Chain Models that take into account the reconstruction time of the systems.

1 Introduction

Distributed or Peer-to-Peer (P2P) storage systems are foreseen as a highly reliable and scalable solution to store digital information [11, 4, 3, 5]. The principle of P2P storage systems is to add redundancy to the data and to spread it onto peers in a network.

There are two classic ways to introduce redundancy: basic replication and erasure codes [13], like the traditional Reed-Solomon (RS) [16]. Many studies compare the reliability of replication against erasure codes [18, 17, 12]. Erasure codes use less additional storage space to obtain the same reliability as replication. On the other hand, replication has the advantage of having no coding/decoding time, of having an easier and faster access to data, and of being adequate in the presence of high churn.

* Partially supported by the INRIA associated team EWIN, by ANR AGAPE, DIMAGREEN, GRATEL, and by Strep EULER.

Furthermore, the reconstruction process of RS systems is costly. In the RS system, the data is divided into small fragments that are sent to different peers. When a fragment of redundancy is lost, the whole original data has to be retrieved to regenerate it. At the opposite, in a system using replication, a repair is done by simply sending again the lost data.

In order to spend less bandwidth in the reconstruction process, the Regenerating Codes were proposed in a recent work [8] as an improvement of the traditional erasure codes. In this coding scheme, the peers that participate of the reconstruction process send, instead of fragments of the data they have, linear combinations of subfragments of the fragments that they keep, in such a way that total transferred data to the newcomer peer is smaller than the original data. This is possible thanks to previous results on Network Coding [1].

In this work, we investigate in detail the use of two hybrid strategies. The first one is usually called Hybrid Coding and was introduced and studied in [17] and [8], respectively. This strategy combines the use of both replication and coding. It tries to get the best of both worlds: the storage efficiency of RS and the repair efficiency of replication. The idea is to keep one full-replica of the data in one peer along with erasure coded fragments spread in the network.

We also propose a new strategy that we name Double Coding in which we improve the idea of Hybrid Coding. Instead of keeping the full-replica of the data in only one peer of the network, we place a copy of each fragment (including the redundant ones) in different peers in the network.

In this paper, we compare Hybrid Coding and Double Coding with RS systems and Regenerating Codes. We study the bandwidth usage of these systems by considering the availability of the peers under the presence of churn, the data durability and the storage space usage. We show that both hybrid strategies perform better than traditional RS systems and that Double Coding is a good option for system developers since it is simple to implement in practice and can perform close to Regenerating Codes in terms of bandwidth usage.

Related Work.

P2P and large scale distributed storage systems have been analyzed by using Markov chains: for erasure codes in [2, 7, 6] and for replication in [14, 5]. In this work, we model Hybrid Coding, Double Coding and Regenerating Codes with Markovian models. We also introduce a new chain for RS systems that models the failure of the reconstructor during a repair.

Rodrigues and Liskov in [17] compare the Hybrid system versus replication in P2P Distributed Hash Tables (DHTs). However, there are no comparisons of the Hybrid system against the traditional erasure codes. Dimakis et al. [8] study the efficiency of bandwidth consumption for different redundancy schemes, among them the Hybrid Coding. They state that the Hybrid Coding has a better availability/bandwidth trade-off than the traditional erasure codes. Both of these works focus on availability and *they do not consider the durability of the data.* They also do not take into account the time to process the reconstructions. By using Markov chains, we exhibit the impact of this parameter on the average system metrics. Furthermore, they only consider RS using an *eager repair* policy,

which is highly inefficient for the bandwidth. In [3], the authors propose the *lazy repair* mechanism to decrease the bandwidth usage in the reconstruction process. Here, we thus compare Hybrid Coding and an RS system using lazy repair.

In [7], Datta and Aberer study analytical models for different lazy repair strategies in order to improve the bandwidth usage under churn. In our work, we employ the lazy repair to minimize the extra-cost in bandwidth even in a system with high availability of peers.

Regenerating Codes [8] is a promising strategy to reduce the bandwidth usage of the reconstruction of the lost data. There are some studies about these codes like in [15], [19], [10] and [9]. However, as far as we know, there is no study of the impact of the reconstruction time in these codes. Most of the results in the literature consider only simultaneous failures. In this work, we introduce a Markovian Model to study the impact of the reconstruction time in Regenerating Codes.

Our Contributions.

- We study the *availability and durability of Hybrid systems*. We compare Hybrid solution with RS system and RC systems.
- We propose a new kind of Hybrid codes, that we refer to as *Double coding*. This new code is more efficient than the Hybrid one. Its performance is close to the one of Regenerating Codes in some cases. Furthermore, explicit deterministic constructions of RC are not known for all sets of parameters. *Double Codes* is then an interesting alternative in this case.
- We model these systems by using Markov chains (Section 3). We derive from these models the *system loss rates* and the *estimated bandwidth usage*. These chains take into account the *reconstruction time* and the more efficient *lazy repair*.
- We analyze different scenarios (Section 4):
  - When storage is the scarce resource, RS system has a higher durability.
  - When bandwidth is the scarce resource, the Hybrid solution is a better option.
- We compare systems for three metrics durability, availability and bandwidth usage for a given storage space, when other studies focus on only two parameters.

In Section 2 we present in detail the studied systems. In the following section we describe the Markov Chain Models used to model these systems. Finally, in Section 4, these systems are compared by an analysis of some estimations on the Markovian models.

2 Description

In distributed storage systems using Reed-Solomon (RS) *erasure codes*, each block of data b is divided into s fragments. Then, r fragments of redundancy are added to b in such a way that any subset of s fragments from the $s+r$ fragments

suffice to reconstruct the whole information of b. These $s+r$ fragments are then stored in different peers of a network. Observe that, the case $s = 1$ corresponds to the simple *replication*. The codes studied in this paper are depicted in Figure 1.

For comparison, we also study *ideal erasure codes* in which there would also be s original fragments and r redundancy fragments spread in the network, but it would be possible to reconstruct a lost fragment by just sending another fragment of information.

The *Hybrid system* is simply a Reed-Solomon erasure code in which one of the $s+r$ peers stores, besides one of the original s fragments of a block b, also a copy of all the other original fragments. This special peer which contains a full copy of b, namely *full-replica*, is denoted by $p_c(b)$.

Following the idea of the Hybrid system, we propose the *Double Coding* strategy. In Double Coding, each of the $s+r$ fragments has a copy in the network. However, differently from the Hybrid approach, we propose to put the copies of the fragments in different peers of the network, instead of concentrating them in a single peer. Consequently, we need twice the storage space of a Reed-Solomon erasure code and also $2(s+r)$ peers in the network. We show, in Section 4, that Double Coding performs much better than RS systems in terms of bandwidth usage and probability to lose data and this disadvantage on storage space is worthy.

Finally, in the Regenerating Codes the original data is also divided into $s+r$ fragments and the fragments are also spread into different peers of a network. However, the size of a fragment in these codes depend on two parameters: the *piece expansion index* i and the repair degree d, as explained in [10]. These parameters are integer values such that $0 \leq i \leq s-1$ and $s \leq d \leq s+r-1$. Given these parameters, the size of a fragment in a Regenerating Code with parameters (s, r, i, d) is equal to $p(d, i)s$ where

$$p(d,i) = \frac{2(d-s+i+1)}{2s(d-s+1)+i(2s-i-1)}.$$

The repair degree d is the number of peers that are *required* to reconstruct a lost fragment. This parameter also impacts the required bandwidth usage to repair a fragment that was lost as we discuss in the next section.

2.1 Reconstruction Process

To ensure fault tolerance, storage systems must have a maintenance layer that keeps enough available redundancy fragments for each block b. In this section, we describe how the lost fragments must be repaired by this maintenance layer in each system.

Reed-Solomon. As stated before, in a Reed-Solomon system the reconstructor $p(b)$ of a block b must download s fragments in the system, in order to rebuild b, before sending the missing fragments to new peers. Most of the works in

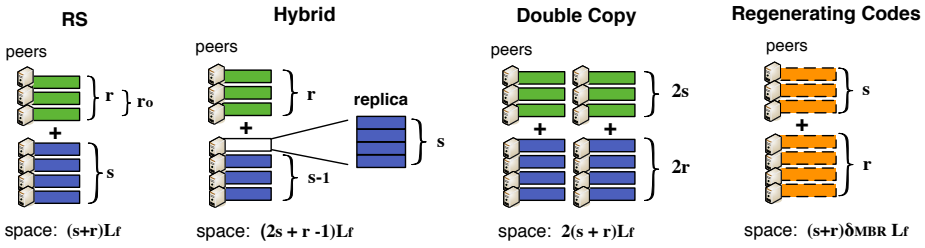


Fig. 1. Description of the redundancy schemes.

the literature consider only the case of the *eager reconstruction*, i.e., as soon as a fragment of data is lost the reconstruction process must start. This is highly inefficient in terms of bandwidth usage, because, in most of the cases, s fragments are sent in the network in order to rebuild only one lost fragment.

Here, we assume that the reconstruction process in a RS system uses the *lazy repair* strategy [7], which can be much more efficient in terms of bandwidth usage. Given a threshold $0 \leq r_0 < r$, the reconstruction process starts only when the number of fragments of b is less than or equal to $s + r_0$. Observe that the case $r_0 = r - 1$ corresponds to the eager reconstruction. Recall that decreasing the value of r_0 correspond to increase the probability to lose the block, i.e., to lose at least $r + 1$ fragments.

When the reconstruction starts, a peer $p(b)$ is chosen to be the *reconstructor*. Note that, when reconstructing the missing redundancy of b, the peer $p(b)$ possesses a full-replica of the block which is discarded afterwards.

Hybrid Coding. In the Hybrid system, recall that $p_c(b)$ is the peer that contains a full-replica of the block b, hence for each block there are $2s + r - 1$ fragments present in the system. When there is a failure, if the peer $p_c(b)$ is still alive, it generates the lost fragments from its full-replica. It then sends the missing fragments to different peers in the network. To be able to do that, the peer only needs to store the initial block or, equivalently, s fragments. As a matter of fact, it can quickly create the other fragments at will.

When the peer $p_c(b)$ fails, a new peer is chosen to maintain the full-replica. In this case, the whole block needs to be reconstructed. This is accomplished by using the traditional Reed-Solomon process, with the addition that the reconstructor keeps a full-replica of the block at the end of the process. From that we see that a Hybrid system can be easily built in practice from an RS system.

Double Coding. Recall that in the Double Coding, for each block there are $2(s + r)$ fragments present in the system. An interesting property of Double Coding is that it keeps the idea of Hybrid Coding, because when a fragment f is lost it is just necessary to ask the peer that contains the other copy of f to send a copy of it to another peer in the network.

Moreover, we can just say that a fragment f is lost in the system if its two copies are lost. In this case, it is necessary to use the Reed-Solomon reconstruc-

tion to rebuild at least one of the copies of f. Since this is an expensive process in terms of bandwidth usage, we also adopt a threshold value $0 \leq r_0 < r$ to let this process more efficient. When $r - r_0$ pairs of the same fragments are lost, a peer $p(b)$ is chosen to be the responsible for downloading s disjoint fragments of the system, rebuilding the block b and the r redundant fragments and resending *only* the *first* copies of the fragments that have lost both of their copies. Then, the *second* copies are sent by the peers that contain the first one.

Regenerating Codes. In these codes there is not the figure of the reconstructor. When a fragment f is lost, a peer that is usually called *newcomer* is in charge of downloading linear combinations of subfragments of the block from exactly d peers in the network in order to replace f.

The amount of information that the newcomer needs to download is equal to $d.\delta(d,i).s$ where

$$\delta(d,i) = \frac{2}{2s(d-s+1) + i(2s-i-1)}.$$

Recall that d peers are required in the reconstruction process. If there are no d peers available in the beginning of the reconstruction process, but there are still s peers on-line, the reconstruction can be still processed by downloading s complete fragments and reconstructing the original information of b as it happens in a RS system.

There are two special cases of Regenerating Codes: the Minimum Bandwidth Regenerating (MBR) codes and the Minimum Storage Regenerating (MSR) codes. The MBR codes correspond to the case in which $i = s - 1$ and in the MSR ones $i = 0$.

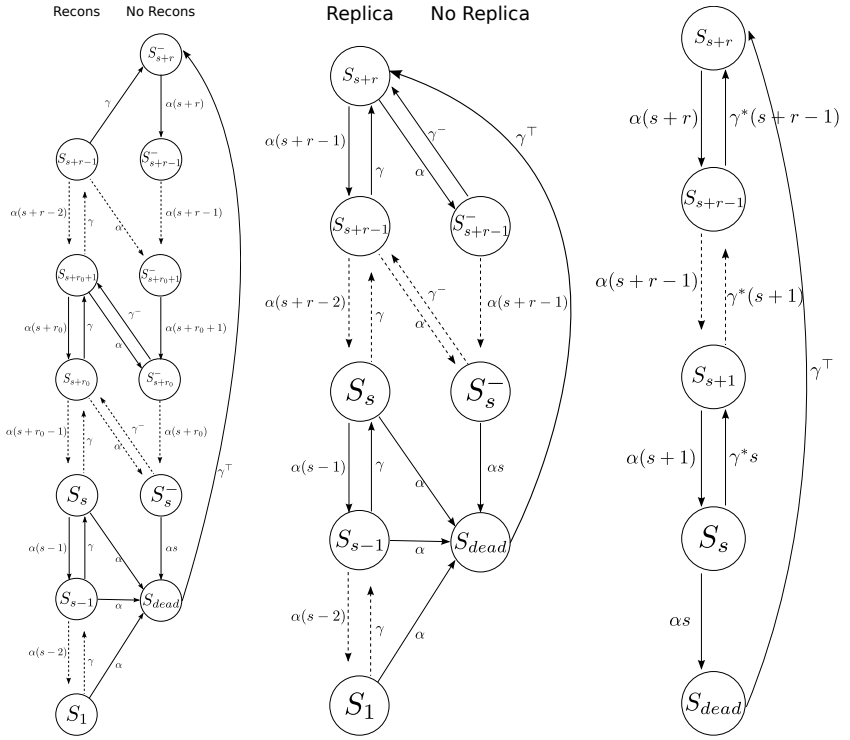
Since the most expensive resource in a network is arguably the bandwidth we use the MBR Regenerating Codes. Observe that these systems have a storage overhead factor δ of $\frac{2d}{2d-s+1}$. That is, each block has $s + r$ fragments, as the RS system, but these fragments are bigger by a overhead factor δ.

In the following section, we present the Markov Chain Models that we use to study the bandwidth usage and the durability of each system.

3 Markov Chain Models

We model the behavior of a block of data in all the cited systems by Continuous Time Markov Chains (CTMCs). From the stability equations of these chains, we derive the bandwidth usage and the system durability.

Model of the Reed-Solomon System. We model the behavior of a block b in a lazy RS system by a CTMC, depicted in Figure 2(a). We did not use the chains classically used in the literature [2, 7]. Our chain models the possible loss of the reconstructor $p(b)$ during a reconstruction. In brief, the states of the chain are grouped into two columns. The level in a column represents the number of Reed-Solomon fragments present in the system. The column codes the presence of the reconstructor $p(b)$: present for the left states and absent for the right ones.



(a) Reed-Solomon. (b) Hybrid system. (c) Regenerating Codes.

Fig. 2. Markov Chain models for different codes.

Fig. 3. Summary of the notations

s	Number of initial fragments
r	Number of redundancy fragments
r_0	Reconstruction threshold
α	Peer failure rate
MTTF	Mean Time To Failure: $1/\alpha$
a	Peer availability rate
d	Number of available peers to reconstruct (RC)
θ	Average time to send one fragment
γ	Fragment reconstruction rate in Hybrid approaches: $\gamma = 1/\theta$
θ^-	Average time to retrieve the whole block
γ^-	Block reconstruction rate: $\gamma^- = 1/\theta^-$
θ^*	Average time to retrieve a d subfragments in RC
γ^*	Fragment reconstruction rate in RC: $\gamma^* = 1/\theta^*$
$\theta^\top$	Average time to reinsert a dead block in the system
$\gamma^\top$	Dead block reinsertion rate

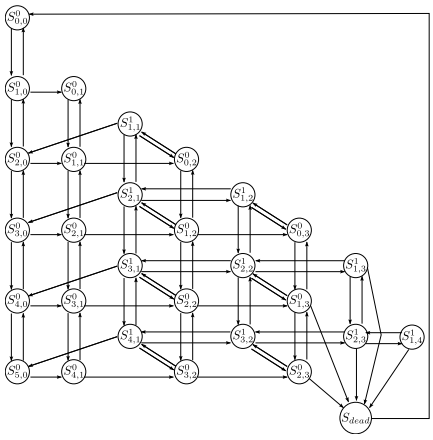


Fig. 4. Markov Chain for Double Coding system for $s = 2$, $r = 3$ and $r_0 = 0$.

Model of the Hybrid System. In Figure 2(b), it is presented the Markov chain that models the behavior of a block b in the Hybrid system. Recall that, in a Hybrid system, $s + r$ Reed-Solomon fragments and one replica are present inside the system. We draw our inspiration from the chain representing the RS system. We code here the presence of the peer $p_c(b)$ in the system, in a similar way to how we code the presence of the reconstructor $p(b)$ in the RS system.

Model of the Double Coded System. We also model the behavior of a block in this system by a continuous-time Markov chain (see Figure 4) to estimate the loss rate of a block and the expected bandwidth usage in the steady state of the system.

Model of Regenerating Codes. Basically, the only difference between the Markov chain that we used to model the RS system and the one that we introduce in this section for Regenerating Codes (see Figure 2(c)) is that in RC-based systems, we do not have the reconstructor. When a fragment is lost, the newcomer will just download linear combinations of subfragments of the other peers that are present in the system.

Model of Ideal Codes. For the ideal system, the chain is similar to the one that we present for Regenerating Codes, only the estimation of bandwidth usage is different.

4 Results

We now use the Markov chains presented in Section 3 to compare the systems we described from the point of view of *data availability, durability and loss rate.*

The bandwidth usage and loss rate plots are estimations from the chains. To estimate the bandwidth usage, we just observe, in the steady state of the chain, the rate that some data in the reconstruction process is transferred times the amount of transferred data. The loss rate is simply the probability to be in the *dead* state in the stationary distribution.

In Subsections 4.1, 4.2 and 4.3, the plots concerning to Regenerating Codes (RC) are estimations taken from the chain where the bandwidth usage is calculated in an *optimal* way, i.e., the estimation considers that the system is a MBR code and, moreover, *all the available peers participate of the reconstruction process.*

Value of the parameters. In the following experiments, we use a set of default parameters for the sake of consistency (except when explicitly stated). We study a system with $N = 10000$ peers. Each of them contributes with $d = 64$ GB of data (total of 640 TB). We choose a system block size of $L_b = 4$ MB, $s = 16$, giving $L_f = L_b/s = 256$ KB. The system wide number of blocks is then $B = 1.6{\cdot}10^8$. The $MTTF$ of peers is set to one year. The disk failure rate follows as $\alpha = 1/MTTF$. The block average reconstruction time is $\theta = \theta^- = \theta^* = \theta^\top = 12$ hours.

Except in the first studied scenario, the availability rate a is chosen to be 0.91 which is exactly the one of PlanetLab [8].

4.1 Systems with same Availability

The first scenario we study is the one we compare the bandwidth usage and the loss rate of the described systems when they have approximately the same availability. Since Ideal, RS and RC systems have the same formula to estimate the availability of each system, they are taken as basis to the hybrid approaches.

In this experiment, we keep the value s constant for all the systems and we increase the availability rate a. For each value of a, we compute the availability for Ideal, RS and RC and, then, we find the value of r for Hybrid coding and also for Double coding that provides the closest value of availability to the one found to Ideal, RS and RC. This experiment provides the results in Figure 5.

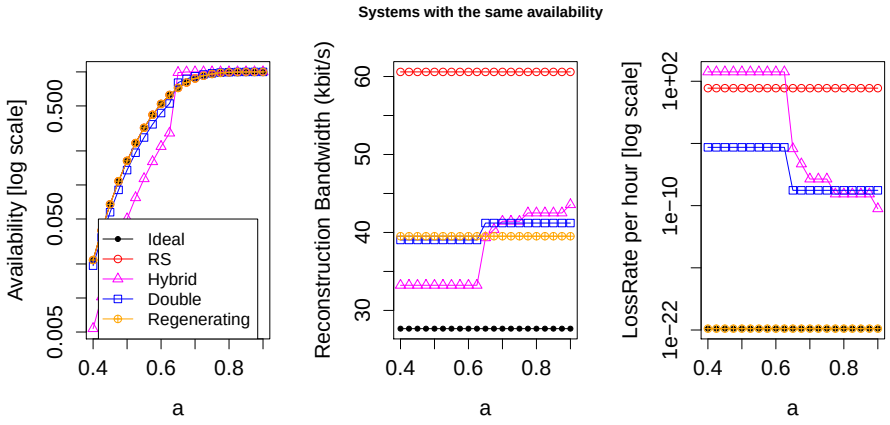


Fig. 5. Systems with same availability.

Since the RS system uses much more bandwidth than the others studied systems, we choose $r_0 = 1$ to provide a lower bandwidth usage. However, one may observe the impact of this choice in the loss rate of this RS system. The Double coding plot has the eager reconstruction strategy, i.e., $r_0 = r - 1$.

Recall that these systems do not use the same storage space, as explained in Section 2. Observe that the hybrid approaches perform as good as regenerating codes in this case. However, the system loss rate is smaller in the regenerating codes.

4.2 Systems with same Durability

In the following experiment, we increase the value of r of an RC system with $s = 16$ and, for each value, the estimation of the system loss rate is taken as a parameter for the others systems.

Given the system loss rate of the RS system, for each other system, the best value of r is considered in order to plot the values of availability and bandwidth usage, i.e., the value of r whose loss rate estimation is the closest to the one of the regenerating code.

In Figure 6, RS and Double coding are both considered to be in the eager case, i.e., $r_0 = r - 1$.

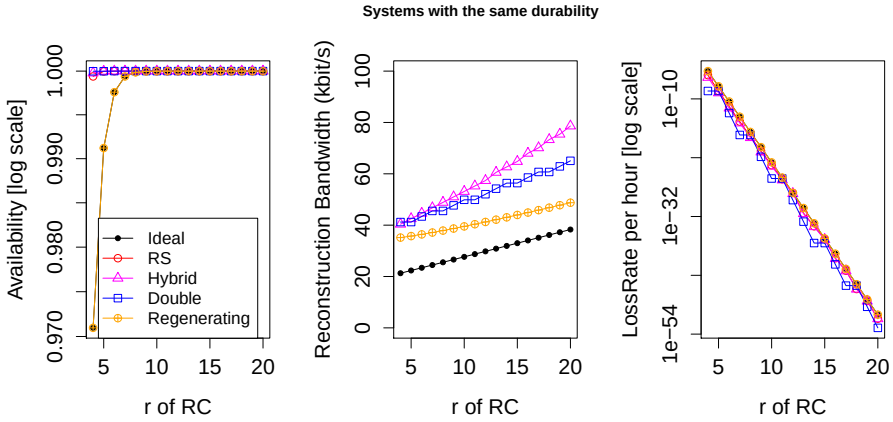


Fig. 6. Systems with same durability.

First, remark that the bandwidth curve of the RS system is not present in the plot since, as commented before, the bandwidth usage in the eager case is much bigger than the bandwidth used by the other systems.

Again, we observe that the hybrid strategies perform well in terms of bandwidth usage when the compared systems have approximately the same loss rate. Recall that these systems do not use the same storage space.

4.3 Systems with same Storage Space

Finally, we compare all the systems when they use the same storage space. The RS system is taken as reference and, then, the redundancy of the others systems is set to use only the space of r fragments of the RS system. Recall that the encoded fragments of regenerating codes are bigger than the RS according to the function presented in Section 2. Consequently, even the regenerating codes have less redundancy fragments in this experiment, when compared with the redundancy of the RS system.

The considered RS system has $r_0 = 1$ in order to let the plot of bandwidth usage in the same scale, since the eager policy performs much worse. Again, observe that the system loss rate is affected by this choice.

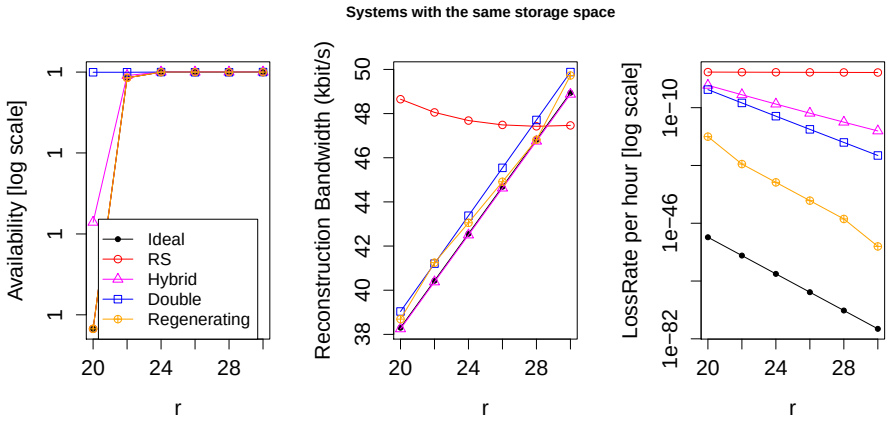


Fig. 7. Systems with same storage space.

Another important remark is that even for systems with the same storage space, the hybrid approaches perform as well as the regenerating codes.

Remember that the last three experiments are based in *Optimal* RC systems, where all the available peers participate of the reconstruction process.

5 Conclusions

In this paper, we studied the *availability and durability of Hybrid systems*. We proposed a new kind of Hybrid codes, namely *Double coding*. Then, we compared Hybrid solutions with Reed-Solomon and Regenerating Codes systems.

We modeled these systems by using Markov chains and derived from these models the *system loss rates* and the *estimated bandwidth usage*. Differently from other studies, these chains take into account the *reconstruction time* of a data-block and the use of the more efficient *lazy repair* procedure. We compared these systems for three metrics: durability, availability and bandwidth usage for a given storage space, when other studies focus on only two parameters. We analyzed different scenarios: when the scarce resource is the storage space or when it is bandwidth.

Double Coding is most of the time more efficient than the Hybrid one. Its performance is close to the one of the best theoretical Regenerating Codes in some scenarios. If Reed-Solomon systems have a higher durability when bandwidth is not limited, Double Coding is a better option when it is a scarse resource.

References

1. R. Ahlswede, N. Cai, S. yen Robert Li, R. W. Yeung, S. Member, and S. Member. Network information flow. *IEEE Transactions on Information Theory*, 46:1204–1216, 2000.

2. S. Alouf, A. Dandoush, and P. Nain. Performance analysis of peer-to-peer storage systems. *Internation Teletraffic Congress (ITC), LNCS 4516*, 4516:642–653, 2007.
3. R. Bhagwan, K. Tati, Y.-C. Cheng, S. Savage, and G. M. Voelker. Total recall: system support for automated availability management. In *Proceedings of Usenix NSDI*, pages 25–25, Berkeley, CA, USA, 2004.
4. W. J. Bolosky, J. R. Douceur, D. Ely, and M. Theimer. Feasibility of a serverless distributed file system deployed on an existing set of desktop pcs. *SIGMETRICS Performance Evalualtion Rev.*, 28(1):34–43, 2000.
5. B.-G. Chun, F. Dabek, A. Haeberlen, E. Sit, H. Weatherspoon, M. F. Kaashoek, J. Kubiatowicz, and R. Morris. Efficient replica maintenance for distributed storage systems. In *Proceedings of Usenix NSDI*, pages 45–58, Berkeley, USA, 2006.
6. O. Dalle, F. Giroire, J. Monteiro, and S. Pérennes. Analysis of failure correlation impact on peer-to-peer storage systems. In *Proceedings of IEEE P2P*, pages 184–193, Sep 2009.
7. A. Datta and K. Aberer. Internet-scale storage systems under churn – a study of the steady-state using markov models. In *Proceedings of IEEE P2P*, volume 0, pages 133–144. IEEE Computer Society, 2006.
8. A. Dimakis, P. Godfrey, M. Wainwright, and K. Ramchandran. Network coding for distributed storage systems. In *Proceedings of IEEE INFOCOM*, pages 2000–2008, May 2007.
9. A. G. Dimakis, K. Ramchandran, Y. Wu, and C. Suh. A survey on network codes for distributed storage. *CoRR*, abs/1004.4438, 2010.
10. A. Duminuco and E. Biersack. A practical study of regenerating codes for peer-to-peer backup systems. In *ICDCS '09: Proceedings of the 2009 29th IEEE International Conference on Distributed Computing Systems*, pages 376–384, Washington, DC, USA, 2009. IEEE Computer Society.
11. J. Kubiatowicz, D. Bindel, Y. Chen, S. Czerwinski, P. Eaton, D. Geels, R. Gummadi, S. Rhea, H. Weatherspoon, C. Wells, et al. OceanStore: an architecture for global-scale persistent storage. *ACM SIGARCH Computer Architecture News*, 28(5):190–201, 2000.
12. W. Lin, D. Chiu, and Y. Lee. Erasure code replication revisited. In *Proceedings of IEEE P2P*, pages 90–97, 2004.
13. M. O. Rabin. Efficient dispersal of information for security, load balancing, and fault tolerance. *Journal of ACM*, 36(2):335–348, 1989.
14. S. Ramabhadran and J. Pasquale. Analysis of long-running replicated systems. In *Proceedings of IEEE INFOCOM)*, pages 1–9, April 2006.
15. K. V. Rashmi, N. B. Shah, P. V. Kumar, and K. Ramchandran. Explicit construction of optimal exact regenerating codes for distributed storage. In *Allerton'09: Proceedings of the 47th annual Allerton conference on Communication, control, and computing*, pages 1243–1249, Piscataway, NJ, USA, 2009. IEEE Press.
16. I. S. Reed and G. Solomon. Polynomial codes over certain finite fields. *Journal of the Society for Industrial and Applied Mathematics*, 8(2):300–304, 1960.
17. R. Rodrigues and B. Liskov. High availability in dhts: Erasure coding vs. replication. In *Peer-to-Peer Systems IV*, pages 226–239. LNCS, 2005.
18. H. Weatherspoon and J. Kubiatowicz. Erasure coding vs. replication: A quantitative comparison. In *Proceedings of IPTPS*, pages 328–338, 2002.
19. Y. Wu, R. Dimakis, and K. Ramchandran. Deterministic regenerating codes for distributed storage. In *Allerton'09: Proceedings of the 47th annual Allerton conference on Communication, control, and computing*, Piscataway, NJ, USA, 2009. IEEE Press.