

## MIT Open Access Articles

*Durable network coded distributed storage*

The MIT Faculty has made this article openly available. **Please share** how this access benefits you. Your story matters.

**Citation:** Abdrashitov, Vitaly, and Muriel Medard. "Durable Network Coded Distributed Storage." 2015 53rd Annual Allerton Conference on Communication, Control, and Computing (Allerton), 29 September-2 October 2015, Monticello, Illinois, IEEE, 2015, pp. 851–56.

**As Published:** <http://dx.doi.org/10.1109/ALLERTON.2015.7447095>

**Publisher:** Institute of Electrical and Electronics Engineers (IEEE)

**Persistent URL:** <http://hdl.handle.net/1721.1/113824>

**Version:** Author's final manuscript: final author's manuscript post peer review, without publisher's formatting or copy editing

**Terms of use:** Creative Commons Attribution-Noncommercial-Share Alike



# Durable Network Coded Distributed Storage

Vitaly Abdrashitov, Muriel Médard  
Dep. of Electrical Eng. & Computer Science  
Massachusetts Institute of Technology  
Cambridge, MA, USA  
Email: {vit,medard}@mit.edu

**Abstract**—In distributed cloud storages fault tolerance is achieved by regenerating the lost data from the surviving clouds. Recent studies suggest using maximum distance separable (MDS) network codes in cloud storage systems to allow efficient and reliable recovery after node faults. MDS codes are designed to use a substantial number of repair nodes and rely on centralized management and a static fully connected network between the nodes. However, in highly dynamic environments, like edge caching in communication networks or peer-to-peer networks, the nodes and the communication links availability is very volatile. In these scenarios MDS codes functionality is limited. In this paper we study a non-MDS network coded approach, which operates in a decentralized manner and requires a small number of repair nodes for node recovery. We investigate long-term behavior of the modeled system and demonstrate, analytically and numerically, the durability gains over uncoded storage.

## I. INTRODUCTION

Modern large-scale distributed cloud storage systems predominantly use coding to tolerate inevitable errors and failures of storage media in order to maintain high availability and avoid data loss. The storage nodes can experience temporary or permanent faults, when the stored information cannot be accessed, due to hardware failure, network connectivity problems, unavailability of a peer in P2P network, etc. When a new node is plugged into the cloud, whether to replace a failed node or to increase redundancy, it needs to be filled with data generated from the other cloud nodes. Reed-Solomon codes [1] have been traditionally used in Redundant Array of Independent Disks (RAID) systems for recovery of the replacement nodes. They belong to the class of maximum distance separable (MDS) codes. Network codes [2], and specifically random linear network codes (RLNC) [3] have been used to construct a MDS regenerating code, which is shown to achieve optimal recovery traffic and storage cost for certain scenarios [4].

MDS-based approaches, however, require a large number of repair nodes and a relatively static cloud structure. This is impractical for highly dynamic conditions like Peer-Aided Content Delivery communication networks, or cloud centers with "hot" highly demanded hard drives often becoming unavailable.

[5] proposed a non-MDS RLNC regenerating approach for distributed and highly dynamic storage systems with little operational planning, and demonstrated its reliability in a simulator-based environment. In this paper we expand

upon [5] to build a model of a RLNC cloud storage with random node losses and recovery from a small number of repair nodes. We analytically investigate its performance, and estimate how many cycles of node failure the storage can tolerate in various scenarios, and, from this point of view, compare a coded system with an uncoded one.

We shortly describe network coded storage basics, introduce the system setup, and provide an algebraic model of the system operation in Section II. Based on that model we derive analytical approximations of the system evolution, and estimates of the storage life time for a small number of repair nodes in Section III. A numerical evaluation of the model and its comparison against the approximations are presented and discussed in Section IV.

## II. PRELIMINARIES AND SYSTEM MODEL

In a random linear network coded [3] storage the original data is cut into  $m$  chunks (segments) of the same sizes. Each chunk is then treated as a huge vector with entries in a finite field. Every storage container is filled with a coded chunk — a random linear combination of all these vectors with coefficient from the finite field. The corresponding vector of  $m$  combination coefficients is also stored on each container and constitutes the code.

During a *recoding* operation several coded chunks can be summed together with random coefficients to create a new coded chunk. The coefficient vector of the new chunk is a linear combination of the coefficient vectors of the old chunks taken with the new random coefficients.

In order to *decode* the original data one needs to read *any*  $m$  containers with linearly independent coefficient vectors, and solve a system of linear equations to recover the original data chunks.

### A. Cloud storage setup

Similarly to [5], we assume that the data is contained in  $m$  *source* segments (packets). These  $m$  segments are put into  $n > m$  (possibly RLNC coded) *storage* segments of the same volume, thus, the redundancy factor is  $r = n/m$ . The storage segments are spread afterwards among  $n_c$  nodes. The size of each node is  $s_c$  storage segments, and  $n = s_c n_c$ .

The storage undergoes independent periodic node failures: at every time step random  $n_l$  nodes out of  $n_c$  lose all their data and need to be recovered. In the recovery phase, which

follows the loss phase, the data is regenerated from  $n_r$  random *repair* nodes chosen from  $n_c - n_l$  working nodes. Both failed and repair nodes are picked uniformly. Each one of  $n_r$  recovery nodes sends a (different) recovery set of  $s_r \leq s_c$  segments to each of  $n_l$  failed nodes (possibly, pre-coding over its own  $s_c$  segments). Each failed node receives  $s_r n_r$  packets and post-codes them into  $s_c$  segments to store. The overall recovery traffic is therefore  $s_r n_r n_l$ . The size of the coding coefficients is negligible with respect to the data segments volume and is not taken into account. After filling up the replaced  $n_l$  nodes, the storage is checked for integrity. If the stored data does not contain  $m$  degrees of freedom required to restore the source data, storage failure is reported.

### B. Algebraic model

The relationship between the source data packets and the storage segments at time  $t \in \{0, 1, 2, \dots\}$  is represented by a  $n \times m$  storage matrix  $M_t \in F^{n \times m}$  with entries in a field  $F$ . The entry  $M[i, j]$  gives the network coding coefficient of  $j$ -th source packet in the  $i$ -th storage segment. For the storage without coding every row of  $M_t$  is a row of identity matrix  $I_{m \times m}$ . We shall interchangeably call the  $i$ -th row  $M[i, \cdot]$  as a vector, a packet, or a segment in the  $k$ -th node for  $k = \lceil i/s_c \rceil$ . Let  $I_{i \times j}$  be the matrix made from the first  $i$  rows ( $i \leq j$ ) or the first  $j$  columns ( $i \geq j$ ) of the identity matrix of dimension  $\max\{i, j\}$ , let  $\mathbf{0}_{i \times j}$  be a  $i \times j$  zero matrix, let  $[n..N]$  be the range of integers between  $n$  and  $N$  inclusively.

The initial storage state is given by a matrix  $M_0$  of full rank ( $m$ ). Each cycle of loss and recovery amounts to left multiplication by a random *evolution matrix*  $W_t$ , and the new storage state is given by  $M_{t+1} = W_{t+1}M_t$ , where  $W_t \in F^{n \times n}$ . By  $W_t^{t'} = \prod_{\tau=0}^{t-t'} W_{t-\tau}$  we shall mean the evolution matrix between times  $t'$  and  $t > t'$ , so that  $M_t = W_t^{t'} M_{t'-1}$ . We also let  $W^t \triangleq W_1^t$ .

Since  $M_t$  and  $W_t$  are block matrices, they can also be indexed by the node number. We shall use bold indices for this purpose. Let  $M_t[\mathbf{i}, \cdot]$  be the  $s_c \times m$  block of  $M_t$  that corresponds to the node  $i$ , and let  $W_t[\mathbf{i}, \mathbf{j}]$ , be the  $s_c \times s_c$  block of  $W_t$  that corresponds to the mapping from node  $j$  in  $M_{t-1}$  to node  $i$  in  $M_t$ . We shall refer to  $W_t[\mathbf{i}, \cdot]$ ,  $W_t[\cdot, \mathbf{j}]$  as  $i$ -th row-group, resp.  $j$ -th column-group of block matrix  $W$ .

Let  $\mathcal{S}_l(t), \mathcal{S}_r(t) \subset [1..n_c]$  be disjoint sets of the lost and the recovery nodes at round  $t+1$  of size  $n_l, n_r$ , respectively. The evolution matrix  $W_t$  has a structure shown on Figure 1:

$$W_t[\mathbf{i}, \mathbf{j}] = \begin{cases} I_{s_c \times s_c}, & \text{if } i = j \notin \mathcal{S}_l(t) \\ U_t^{ij}, & \text{if } i \in \mathcal{S}_l(t), j \in \mathcal{S}_r(t) \\ \mathbf{0}_{s_c \times s_c}, & \text{otherwise,} \end{cases} \quad (1)$$

where  $U_t^{ij} = U_t^{ij,post} U_t^{ij,pre}$  is the product of the *pre-coding recovery matrix*  $U_t^{ij,pre} \in F^{s_r \times s_c}$ , which describes the mapping from the packets on repair node  $j$  to the  $s_r$  packets sent to node  $i$ , and the *post-coding recovery matrix*

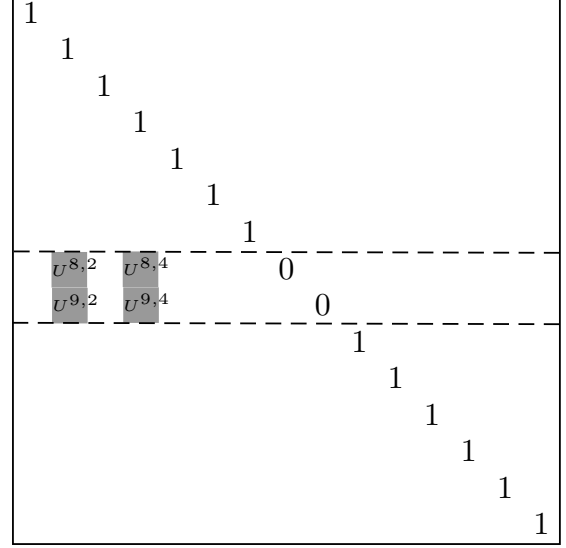


Fig. 1. Evolution matrix  $W_t$  structure.  $\mathcal{S}_l(t) = \{8, 9\}$ ,  $\mathcal{S}_r(t) = \{2, 4\}$ ,  $s_c = 1$ .

$U_t^{ij,post} \in F^{s_c \times s_r}$ , which describes the mapping from the packets received by node  $i$  from node  $j$  to the regenerated packets to be stored on node  $i$ . Note that  $\text{rank } W_t = n - n_l s_c$ .

For a network coded storage we assume that each entry of  $U_t^{ij,pre}$ ,  $U_t^{ij,post}$  is drawn randomly independently uniformly from non-zero elements of  $F$ . For a storage without coding  $U_t^{ij,pre}$ ,  $U_t^{ij,post}$  consists of rows and columns of  $I_{s_c \times s_c}$ , respectively.

As the storage  $M_t$  evolves with time,  $\text{rank } M_t$  decreases or remains the same. For  $|F| \gg n \geq m$  the original  $m$  segments' random storage coefficients in  $M_0$  with high probability are such that any  $m$  coded packets (rows) of  $M_0$  are independent, and, therefore,

$$\begin{aligned} \text{rank } M_t &= \min\{\text{rank } W^t, \text{rank } M_0\} \\ &= \min\{\text{rank } W^t, m\}. \end{aligned} \quad (2)$$

$m$  linearly independent packets are necessary and sufficient to recover the source packets from the storage. Let  $T_{life}$  be the time to storage failure, i.e the time when  $\text{rank } W_1^t$ , and, hence,  $\text{rank } M_t$  becomes less than  $m$  for the first time

$$T_{life} \triangleq \min\{t : \text{rank } M_t < m\}. \quad (3)$$

Our goal is to estimate average  $T_{life}$  for various recovery strategies and system parameters.

### III. ANALYSIS

$T_{life}$  is the hitting time of a certain subset of states of discrete Markov process  $M_t$  with an enormously large state space. We shall analyze this process via approximating it with a low-dimensional Markov chain.

Note that if recovery traffic  $n_r n_l s_r$  is less than the number of source packets  $m$ , then there is a chance of using the same  $n_r$  repair nodes to regenerate the data on the other  $n_c - n_r$  nodes and lower the rank of  $M_t$  to  $n_r s_c$ . If  $n_r s_c \leq m - s_c$ ,

this sequence of losses and repairs would break the storage. Hence, theoretically  $T_{life}$  can take very small values, and no code can guarantee integrity of the storage even after  $\frac{n_c - n_r}{n_l}$  iterations:

$$\Pr[T_{life} \leq \lceil \frac{n_c - n_r}{n_l} \rceil] > 0. \quad (4)$$

Additionally, the time till the first encounter of this sequence, firstly, follows a geometric distribution with finite mean and, secondly, upperbounds  $T_{life}$ . Therefore, almost surely

$$\mathbb{E}[T_{life}] < \infty, \forall m \geq (n_r + 1)s_c. \quad (5)$$

Let  $T > 0$  and let

$$\widetilde{W}^t \triangleq \widetilde{W}_{T-t+1}^T = \widetilde{W}_T \widetilde{W}_{T-1} \cdots \widetilde{W}_{T-t+1},$$

where  $\widetilde{W}_t[i, j] = W_t[i, j]$  for all entries, except for the recovery submatrix, whose entries  $\{\widetilde{W}_t[i, j]\}_{(i, j) \in S_l(t) \times S_r(t)}$  are drawn uniformly at random from real interval  $(0, 1)$ . Correspondingly,  $\widetilde{W}^t \in \mathbb{R}^{n \times n}$ . Consider  $N_0(t) = |\{j : \widetilde{W}^t[i, j]_c = \mathbf{0}_{s_c \times s_c} \forall i\}|$  for  $t \in [0..T]$  which is the number of zero column-groups of size  $s_c$  in  $\widetilde{W}^t$ . Assume that  $N_0(0) = 0$ . Clearly,  $N_0(t) \in [0..n_c]$ .

**Theorem 1.** *With probability 1 the rank of the evolution matrix is bounded by*

$$\text{rank } W^T \leq s_c(n_c - \max_{t \leq T} N_0(t)). \quad (6)$$

$N_0(t)$  follows the Markov property with  $\Pr[N_0(t+1) = j | N_0(t) = i] = p_{i,j}$ , where

$$p_{i,j} = 1_{\{i=j\}} \text{Hg}_{n_l/n_c}^{n_l/i} + \sum_{\substack{k \in [0..n_r] \\ l \in [1..n_l] \\ l-k=j-i}} \text{Hg}_{n_r/n_c}^{k/i} \text{Hg}_{n_l/n_c - n_r}^{l/n_c - i - (n_r - k)}, \quad (7)$$

where  $\text{Hg}_{n/N}^{k/K} = \frac{\binom{K}{k} \binom{N-K}{n-k}}{\binom{N}{n}}$  is the pmf of the hypergeometric distribution with  $n$  trials,  $N$  items, and  $K$  possible successes.

*Proof.* Each unit of  $N_0(t)$  corresponds to  $s_c$  zero columns of  $\widetilde{W}^t$  and to at least  $s_c$  units of the nullity of  $\widetilde{W}^t$ . Therefore,  $\text{rank } \widetilde{W}^t \leq s_c(n_c - N_0(t)) \forall t \leq T$ . Since  $\text{rank } \widetilde{W}^t$  is non-increasing with  $t$ ,  $\text{rank } \widetilde{W}^T \leq \min_{t \leq T} s_c(n_c - N_0(t))$ .

Multiplying  $\widetilde{W}^t$  by  $\widetilde{W}_{T-t}$ , or  $W_{T-t+1}^T$  by  $W_{T-t}$  on the right performs linear operations on the columns over field  $F$  or  $\mathbb{R}$ . Since  $\mathbb{R}$  is an uncountable field with characteristic zero, no linear dependency between the columns of  $\widetilde{W}^t$  would be invalid in  $W_{T-t+1}^T$  with probability 1 for any  $t$ . Therefore,  $\text{rank } W^T \leq \text{rank } \widetilde{W}^T$ , and bound (6) follows.

Let  $N_0(t) = i$ . As  $\widetilde{W}^t$  is multiplied by  $\widetilde{W}_{T-t}$ , random  $n_l$  column-groups of  $\widetilde{W}^t$ ,  $l$  out of which being non-zero, are chosen to be, first, added with random non-zero coefficients to  $n_r$  other column-groups,  $k$  out of which are zero, and, second, replaced with zeros.

- If all the  $n_l$  column-groups of  $\widetilde{W}^t$  were zero column-groups ( $l = 0$ ), then the multiplication does not change the matrix and  $N_0$ . This happens w.p.  $\text{Hg}_{n_l/n_c}^{n_l/i}$ , which corresponds to the first term in Equation (7).
- Otherwise ( $l > 0$ ), non-zero column-groups are added with positive coefficients to the  $n_r$  column-groups, and they become non-zero in  $\widetilde{W}^{t+1}$ . In this case  $N_0(t+1) = i + l - k \triangleq j$ . Expressing the probabilities of having specific values for  $l$  and  $k$  gives the second term in Equation (7). □

The next theorem derives explicit expressions for hitting times  $T_k \triangleq \min\{t : N_0(t) = k\}$  for the case of single failed and repair node.

**Theorem 2.** *For  $n_r = n_l = 1$ ,  $s_r = s_c$ , and  $|F| \gg n$ , the bound 6 is tight with high probability.*

$$\text{rank } W^T = s_c(n_c - N_0(T)). \quad (8)$$

The average hitting times are given by

$$\mathbb{E}[T_k] = \mathbb{E}[\text{Time when } \text{rank } W^t \text{ reaches } (n_c - k)s_c] = \frac{k(n_c - 1)}{n_c - k}. \quad (9)$$

*Proof.* For  $n_r = n_l = 1$  from Equation (7)

$$p_{i,i+1} = 1 - p_{i,i} = \frac{n_c - i}{n_c} \frac{n_c - i - 1}{n_c - 1}. \quad (10)$$

$N_0(t)$  never decreases, and  $\max_{t \leq T} N_0(t) = N_0(T)$ .

For  $s_r = s_c$  each nonzero block  $W_{T-t}[i, j]_c$  is either  $I_{s_c \times s_c}$ , or can be considered randomly chosen from  $(F \setminus 0)^{s_c \times s_c}$ . A random matrix  $\alpha$  from  $F^{s_c \times s_c}$  has full rank  $s_c$ , with probability at least  $1 - \frac{1}{|F| - 1}$ .

Equation (8) holds for  $t = 0$ . Assume it hold at some  $t \geq 0$ . At each time step  $t + 1$   $W_{T-t+1}^T$  is multiplied by  $W_{T-t}$  to result in  $W_{T-t}^T$ . This replaces column-groups  $A, B$  with  $0, B + A\alpha$ .

- If  $A$  is zero,  $W_{T-t+1}^T = W_{T-t}^T$ , the rank and  $N_0(t)$  do not change, Equation (8) holds at  $t + 1$ .
- If  $B$  is zero,  $A, B = A, 0$  are replaced with  $0, A = B, A$ , the rank and  $N_0(t)$  do not change, Equation (8) holds at  $t + 1$ .
- If  $A, B$  are non-zero,  $B$  is replaced with  $B + A\alpha$ . With high probability  $\alpha$  is full-rank,  $A\alpha$  is independent of  $B$ , and  $B + A\alpha$  is full-rank. The rank decreases by  $\text{rank } A = s_c$  (with high probability),  $N_0(t)$  increases by one, Equation (8) still holds at  $t + 1$ .

By induction with high probability Equation (8) holds for any  $t$ .

Let  $T_{i,k}$  be the average hitting time of state  $k$ , starting from state  $i \leq k$ . We have

$$\begin{aligned} T_{i,k} &= p_{i,i} T_{i,k} + p_{i,i+1} T_{i+1,k} + 1 \\ T_{i,k} &= 1/p_{i,i+1} + T_{i+1,k}, \end{aligned}$$

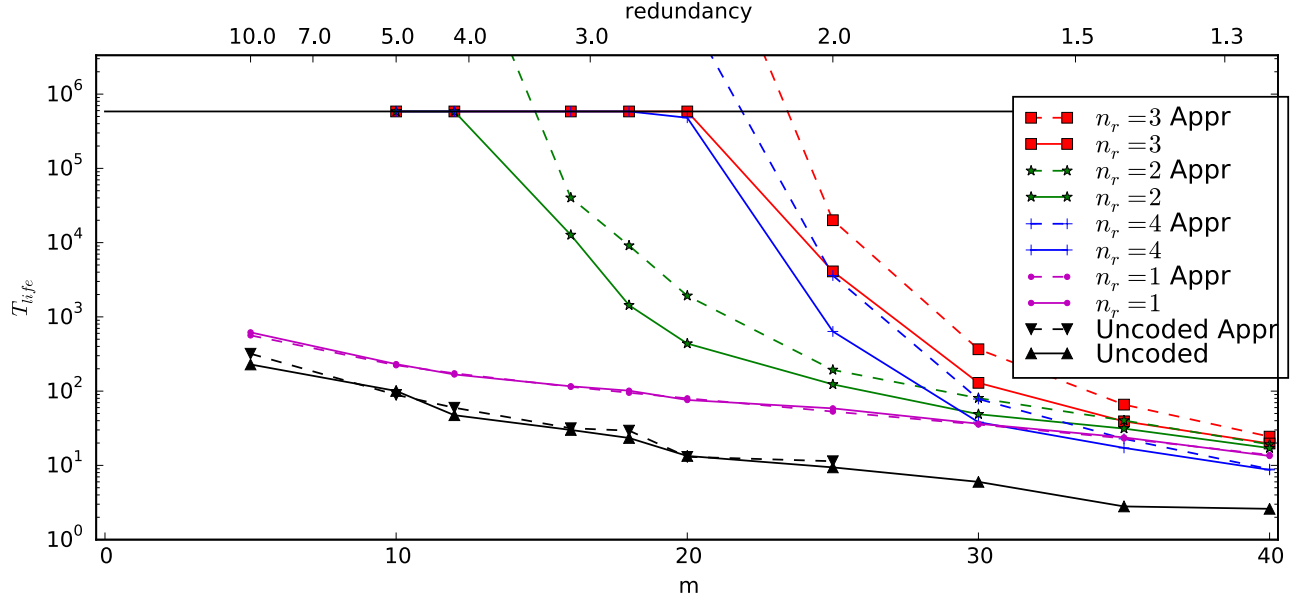


Fig. 2. Simulations and analytical approximations of storage life time.

which in combination with  $T_{k,k} = 0$  and Equation (10) gives

$$\begin{aligned} T_{0,k} &= \mathbb{E}[T_k] = \sum_{i=0}^{k-1} \frac{1}{p_{i,i+1}} = \sum_{i=0}^{k-1} \frac{n_c}{n_c - i} \frac{n_c - 1}{n_c - i - 1} \\ &= n_c(n_c - 1) \left( \frac{1}{n_c - k} - \frac{1}{n_c} \right) = \frac{n_c(n_c - 1)k}{(n_c - k)n_c}, \end{aligned}$$

which proves Equation (9).  $\square$

For this special case Theorem 2 provides an analytical expression for the average storage life time. It is the hitting time of state  $n_c - (m - s_c)$ , which corresponds to rank  $m - s_c$ :

$$\mathbb{E}[T_{life}] = \frac{(n_c - m + s_c)(n_c - 1)}{m - s_c}. \quad (11)$$

The general case upper bound

$$\mathbb{E}[T_{life}] \leq T^+ : \mathbb{E}[\max_{t \leq T^+} N_0(t)] = n_c - m/s_c + 1, \quad (12)$$

which follows from (6), does not have an explicit formula, but it can be calculated numerically from the transition matrix given by (7).

In the scenario without storage coding every row of  $M_t = M_t I_{m \times m}$  is a row of identity matrix  $I_{m \times m}$ , and every storage packet is just a copy of some source packet.  $r$  copies of every source packet are originally put into the storage, producing  $n = rm$  storage segments of  $m$  different kinds. Assume that  $s_c < m$ , and no cloud contains more than one copy of any source packet. Let  $k$ -th family  $\mathcal{F}_k(t) = \{i : M_t[i, k] \neq \mathbf{0}_{s_c \times 1}\}$  be the collection of the indices of the clouds which have the  $k$ -th source segment  $I_{m \times m}[k, \cdot]$ , let  $N_k(t) = |\mathcal{F}_k(t)| \leq n_c$ , and let  $R(t) = \sum_{k=1}^m \mathbf{1}\{N_k(t) > 0\}$  be the number of non-empty

families. Then  $N_k(0) = r$ ,  $R(0) = m$ , and the storage failure condition is  $R(T_{life}^{uncod}) = m - s_c$ . We can derive

$$\mathbb{E}[R(T_{life}^{uncod})] = m \Pr[N_1(T_{life}^{uncod}) > 0] = m - s_c. \quad (13)$$

$N_k(t)$  follows a Markov chain with transition probabilities  $\Pr[N_k(t+1) = j | N_k(t) = i] = q_{i,j}$

$$\begin{aligned} q_{i,i+1} &= q_{i,i-1} = \frac{i}{n_c} \frac{n_c - i}{n_c - 1} \\ q_{i,i} &= 1 - 2q_{i,i+1}. \end{aligned} \quad (14)$$

#### IV. QUANTITATIVE RESULTS

In addition to theoretical analysis, we test performance of the algebraic model, described in Section II-B, in a software simulation and compare it to the analytical results. In each test  $m$  source data segments, randomly generated from  $\mathbb{F}_{65537}^m$ , are encoded into  $n = n_c = 50$  storage packets,  $s_c = 1$  per node, and put into matrix  $M_0$ . At each discrete time step  $t > 0$   $n_l$  uniformly chosen storage packets are erased (node fault), and filled with random linear combinations of  $n_r$  other packets (node recovery), also chosen uniformly at random. We assume  $n_l = 1$  in all tests, except for the test with  $n_r = 4, n_l = 2$ . Additionally, we perform uncoded storage tests, where  $\sim n/m$  copies of each source packet are distributed possibly uniformly among  $n = 50$  nodes, and recovery is performed by copying all the packets from the repair node onto the failed node.

For a range of  $m$ , which also corresponds to a range of redundancy  $r = n/m$  values, we measure  $T_{life}$  (time when rank  $M_t$  becomes  $m - 1$ ) average over 20 independent experiments. The resulting average life times are shown on Figure 2. The maximal simulation time was set  $\approx 6 * 10^5$ . The test storages, that did not reach rank  $m - 1$  by that time, are depicted by a point on the black horizontal line.

The plot also demonstrates the analytical approximations (13) and (11) for the uncoded and coded ( $n_r = 1$ ) scenarios, respectively, and the general case upper bound (12).

In accordance with Equation (11), for single node coded repair the storage life time grows linearly with redundancy. Although the uncoded storage exhibits superlinear behavior, its life time is smaller by a factor 3 to 10 for redundancies 10 and 1.5, respectively. This gain is due to RLNC, which obviates the need to have exact copies of all the original data packets in the storage (a coupon collector problem [6]), any  $m$  independent linear combinations are enough.

In contrast, the test life time of a storage with several repair nodes ( $n_r > 1$ ) grows at least exponentially with redundancy and  $n_r$ . The average gain in  $T_{life}$  over single node repair has factors of 5 for  $n_r = 2$ , and more than 5000 for  $n_r = 3$  when the redundancy is  $r = 2.5$ . When the lost packets are recovered from a single repair node, the coded packets from different nodes are never mixed together in a linear combination. Packet internode recoding creates multiple dependencies between many packets in the storage, and makes it harder to eliminate degrees of freedom.

## V. CONCLUSIONS

We studied a RLNC distributed cloud storage, where the data for the failed nodes is regenerated from few repair nodes in a decentralized manner. We proposed an algebraic model, and used it to analyze durability of the system. The derived analytical approximations of the average storage life time provide qualitative and quantitative intuition for the choice of the system parameters.

## REFERENCES

- [1] I. S. Reed and G. Solomon, "Polynomial codes over certain finite fields," *Journal of the society for industrial and applied mathematics*, vol. 8, no. 2, pp. 300–304, 1960 (cit. on p. 1).
- [2] R. Ahlswede, N. Cai, S.-Y. Li, and R. W. Yeung, "Network information flow," *Information Theory, IEEE Transactions on*, vol. 46, no. 4, pp. 1204–1216, 2000 (cit. on p. 1).
- [3] T. Ho, M. Médard, R. Koetter, *et al.*, "A random linear network coding approach to multicast," *Information Theory, IEEE Transactions on*, vol. 52, no. 10, pp. 4413–4430, 2006 (cit. on p. 1).
- [4] A. G. Dimakis, P. B. Godfrey, Y. Wu, *et al.*, "Network coding for distributed storage systems," *Information Theory, IEEE Transactions on*, vol. 56, no. 9, pp. 4539–4551, 2010 (cit. on p. 1).
- [5] F. H. Fitzek, T. Toth, A. Szabados, *et al.*, "Implementation and performance evaluation of distributed cloud storage solutions using random linear network coding," in *Communications Workshops (ICC), 2014 IEEE International Conference on*, IEEE, 2014, pp. 249–254 (cit. on p. 1).
- [6] S. Deb, M. Médard, and C. Choute, "Algebraic gossip: A network coding approach to optimal multiple rumor mongering," *Information Theory, IEEE Transactions on*, vol. 52, no. 6, pp. 2486–2507, 2006 (cit. on p. 5).