Staying alive — Network coding for data persistence in volatile networks


http://dx.doi.org/10.1109/ACSSC.2016.7869145

Institute of Electrical and Electronics Engineers (IEEE)

Author’s final manuscript

http://hdl.handle.net/1721.1/113670

Creative Commons Attribution-Noncommercial-Share Alike

http://creativecommons.org/licenses/by-nc-sa/4.0/
Staying Alive — network coding for data persistence in volatile networks

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

Abstract—In volatile network environments, node connectivity and availability changes rapidly. This poses a challenge to efficient repair of failed nodes in distributed storage with erasure-correcting code. The commonly considered bandwidth-efficient data regenerating schemes either rely on availability of specific storage nodes in the network, or require a substantial number of repair nodes. We propose a RLNC storage scheme, which can operate well with highly changing network connectivity, and allows data decoding even after many failures and repairs. We demonstrate its substantial performance gain over the existing regenerating schemes.

I. INTRODUCTION

In large-scale distribute storage systems data is spread among multiple individually unreliable nodes. Redundancy is required to ensure data availability and durability. Many practically deployed systems create redundancy through the data replication, when multiple copies of each data block are stored across several storage nodes. However, replication entails a large storage overhead and increase the total storage costs several times.

Erasure coding is an alternative way to provide redundancy with much smaller storage overhead [1]. Traditional erasure codes, such as Reed-Solomon codes, however, are not suitable for large-scale systems, because of prohibitively large node repair costs. Regenerating codes [2] were proposed to solve the problem of large repair bandwidth. Locally repairable codes [3] can provide small repair locality, i.e. the number of helpers, or surviving nodes that are contacted by a replacement newcomer node during repair.

However, in volatile network environments, such as peer-aided content delivery networks or ad-hoc wireless mobile networks, the surviving nodes often become unavailable for repair. Locally repairable codes rely on fixed sets of helpers for each failed node, and do not work in these scenarios. Regenerating codes are designed for the worst-case selection of the helper nodes, and can work, but require a large storage overhead, because the repair locality is low.

In this paper we study Durable Random Network codes (DRNC) — a random linear network coded (RLNC) [4] distributed storage scheme, which can operate well with highly changing network connectivity, and is persistent in a sense that it allows the user data decoding even after many cycles of node failures and repairs. The helper data for node repairs is drawn from small node groups, but no specific node in a group is essential for the repair. The scheme was proposed by the authors in [5]. Via numerical simulations, we demonstrate the DRNC performance with various model parameters and explain the effects observed.

II. SYSTEM MODEL

In this section we describe the proposed model of Durable Regenerating Network codes (DRNC). We will make use of the following notation. \([M,N] = \{M,M+1,\ldots,N\}\) is the range of integers between \(M\) and \(N\). If \(M > N\), \([M,N] = \emptyset\). By \([N]\) we assume \([1,N]\), \((x)^+ = \max\{x,0\}\).

Each symbol is considered an element of a finite field \(F = \mathbb{F}_q\) with \(q\) elements.

A source file \(S \in \mathbb{F}^B\) of size \(B\) (symbols) is encoded into a \(n\alpha\) dimensional vector \(X \in \mathbb{F}^B\). These \(n\alpha\) symbols are then stored across \(n\) nodes, \(\alpha\) symbols per node. Assuming linear encoding, the stored content can be represented as \(X = MS\), where \(M \in \mathbb{F}^{n\alpha \times B}\) is the encoding matrix. When a node fails, its content is considered lost, and a newcomer replacement node is placed into the storage. The replacement node downloads \(\beta\) symbols each from a set of \(d\) other nodes. Those \(\beta\) symbols on each helper node are computed from its \(\alpha\) stored symbols. From the received \(d\beta\) symbols of the helper data, the replacement node computes \(\alpha\) symbols to store.

All aforementioned encoding and recoding operations are performed using random linear network coding (RLNC) [4]. Therefore, our system performs functional node repair, i.e. the content of a repaired node is in general different from the content of the failed node it replaces. The original node symbols \(X_i\) are random linear combinations of source file symbols \(S_1,\ldots,S_B\), i.e. each element of encoding matrix \(M\) is drawn uniformly at random from \(\mathbb{F}\). To generate helper symbols each helper node recodes over its \(\alpha\) symbols, and \(d\beta\) helper symbols are recoded to fill the newcomer node.

The difference between our model and the classical regenerating codes framework [2] comes from average helper nodes selection. At each failure, \(d\) helper nodes are
picked uniformly at random out of \( n - 1 \) surviving nodes, independently of all previous helper choices. We also assume each failed node is chosen uniformly at random out of \( n \) nodes.

Let discrete time \( t \) count the number of complete node failures and repairs. \( t = 0 \) corresponds to the original storage state. Let \( X_t \) be the nodes content after \( t \) node failures and repairs, \( X_0 = X = MS \). Let \( M^t \) be the storage encoding matrix at time \( t \), \( W_t \in \mathbb{F}_{n \alpha}^{n \alpha \times n \alpha} \) be the coding coefficients matrix of \( X_t \) over \( X_{t-1} \), and \( W^t = W_t W_{t-1} \cdots W_1 \) be the coding coefficients matrix of \( X_t \) over \( X_0 \), i.e. \( X_t = M^t S = W_t X_{t-1} \cdots W_1 X = W_t X \), \( M^t = W^t M \).

Data collection is performed by downloading all data from some subset of \( k_d \geq B/\alpha \) symbols. If out of these \( k_d \alpha \) symbols there exists a set of \( B \) symbols whose coding coefficient vectors in \( M_t \) are linearly independent, then decoding of the source file symbols \( S \) is possible via post-multiplying by the inverse of the corresponding \( B \times B \) submatrix of \( M^t \). On the other hand, decoding is impossible for any \( k_d \) if \( \text{rank} \ M^t < B \). In this situation we declare a storage failure and say that the storage is dead. Storage life time \( T_{\text{life}} \) is defined as the number of the node failures and repairs before the storage failure. For \( t \leq T_{\text{life}} \), the storage is considered alive.

### III. Rank Dynamics

Storage failure happens when for the first time \( \text{rank} \ M^t < B = \text{rank} \ M \) (w.h.p.). Consider a linear operator \( T : F^n \rightarrow F^n \), \( T(x) = W^t x \), and let \( m = \dim \ker T = n - \text{rank} \ W^t \). Then

\[
\Pr[\text{rank} \ M^t < B] = \Pr[\text{rank} \ W^t M < \text{rank} \ M] = \Pr[\text{colspan} M \cap \ker T = \emptyset] = 1 - \Pr[M_i \notin \ker T] \Pr[M_i \notin \ker T \cup \{M_1\}] \cdots \\
\approx 1 - \frac{k-1}{q^{m+1-n}} \approx \frac{q^{\min(m+1,n)}}{q^{m+1}} = q^{\min(k - \text{rank} W^t - 1, 0)},
\]

where \( M_i \) is the \( i \)-th column of \( M \). Therefore, for a large field size \( q \), a storage failure is unlikely while \( \text{rank} \ W^t \geq B \).

In order to find the storage life time, it is enough to analyze the dynamics of the rank of \( W^t \).

Per Theorem 1 from [5], the expected rank of \( W^t \) is bounded by

\[
\mathbb{E}[\text{rank} \ W^t] \leq \alpha(n - \mathbb{E}[\max_{t \leq T} N_0(t)]),
\]

where \( N_0(t) \) is a Markov chain with transition probability

\[
\Pr[N_0(t+1) = j | N_0(t) = i] = p_{i,j} = \begin{cases} \frac{i/n + H_{g_{i}/n}^{i/j}}{n-d+1}, & i = j \\ H_{g_{i/d}/n}^{i+1/j} & i \neq j, i - j + 2 \in [d+1] \\ 0, & \text{otherwise} \end{cases}
\]

![Fig. 1. Average rank dynamics for \( n = 20, \alpha = \beta = 2, d = 9 \). DRNC significantly outperforms the traditional regenerating codes in the effective coding rate](image)

(2)

and \( N_0(0) = 0 \), where \( H_{g_{i/n}}^{j/i} = \binom{K}{i}(\binom{N-K}{j-i})^{-1} \) is the pmf of the hypergeometric distribution with \( n \) trials, \( N \) items, and \( K \) possible successes.

Though this upper bound is not tight, it turns out to match the simulated life time within a constant factor for large enough field sizes. The rank decrease is superexponential in \( t \); for rank \( W^t \) to change from \( r \) to \( r - 1 \), it takes on average \( O(e^{\beta(t)}) \) node failures, where \( f(r) \) is superlinear (Figure 1).

For many practical purposes this life times are larger than the time \( T_{\text{obs}} \) it takes for the stored data to become obsolete. We can talk about the effective storage coding rate \( R_{\text{eff}} \) with confidence \( p \), if \( \Pr[\text{rank} \ W_{t+1}^t > \text{rank} \ W_t^t > R_{\text{eff}}] \geq p \), where probability is taken over random failed and helper nodes selection. Such \( R_{\text{eff}} \) is of the order of \( E[\text{rank} \ W_{t+1}^t] \). E.g. \( R_{\text{eff}} \approx 0.3 \) \( E[\text{rank} W^{2000}] \) with \( p = 0.997 \) for the scenario on Figure 2, and \( R_{\text{eff}} \approx E[\text{rank} W_{t+1}^t] \) for \( p = 0.5 \).

Figure 1 compares the effective coding rate of DRNC against two extreme alternative schemes in terms of the helper selection. The traditional Regenerating codes (RC) [2] provide protection against the worst-case selection of failed and helper nodes and allow decoding from an arbitrary set of \( k \) nodes. They achieve coding rate

\[
R_{RC} = \frac{1}{n\alpha} \sum_{i=0}^{k-1} \min\{\alpha, (d - i)^+ \beta\} \leq d/n,
\]

where the last bound can be achieved at the minimum-storage regeneration (MSR) point with \( \beta = \alpha/(d-k+1) \) and \( k = d \).

The opposite scenario of fixed helper sets for every possible node failure has been studied in the framework of Locally Repairable codes (LRC) [3]. Coding rate with \( d \) helper nodes is bounded by

\[
R_{LRC} \leq \frac{d}{d + 1},
\]

where the last bound can be achieved at the minimum-storage regeneration (MSR) point with \( \beta = \alpha/(d-k+1) \) and \( k = d \).

The rate is maximized for \( d + 1 | n \) by a code where every \((d + 1)\)-th node is the parity check of the previous \( d \) nodes, and a failed node is repaired by contacting the fixed set of \( d \) other nodes in its parity relation.
As shown on Figure 1, DRNC typically significantly outperforms RC in terms of effective coding rate.

IV. QUANTITATIVE RESULTS

In this section we explore the effects of the parameter variation on the model performance. We numerically simulate the random evolution of $W^t$ over finite field $\mathbb{F}_{65537}$ (when not indicated otherwise), and compute its average rank.

A. Decoding probability

![Graph showing decoding probability](image1)

Fig. 2. Average data decoding probability from random $k_d$ nodes for $n = 20, \alpha = \beta = 2, d = 9, B = 28$.

Figure 2 demonstrates the evolution of the probability $P_{k_d}(t)$ that a random set of $k_d \geq B/\alpha$ nodes contain enough information to decode the source file. $P_n(t)$ gives the probability that the storage is alive at time $t$. For file size $B = 28 < E[\text{rank } W^{2000}]$, $n = 20, \alpha = \beta = 2, d = 9$, and $t = 2000$ the plots show that for $k_d = B/\alpha$ $P_{B/\alpha} \approx 0.99$, and for higher $k_d$ the decoding probability is essentially the same as $P_n$. Although the MDS property of decoding from any set of $B/\alpha$ nodes is not guaranteed when the storage is alive, it is feasible with high probability $P_{B/\alpha}/P_n \approx 0.993$.

B. Effect of field size

![Graph showing rank dynamics](image2)

Fig. 3. Average rank dynamics for various field sizes and $n = 20, \alpha = \beta = 2, d = 10$. Negligible difference for $q \geq 257$.

For a large enough field size the exact $\text{rank } W^t$ dynamics is described by a random evolution of a certain matroid in the sense that the specific values of the random coding coefficients are irrelevant for the rank. Linear dependencies between the nodes result only from the nature of the repair process and not from finiteness of the field size.

However, if the field size is too small, additional linear dependencies are likely to appear, and the rank goes down faster. Figure 3 measures the effect of the size of the underlying finite field. The extra rank decrease is significant for $q = 2, 3$, noticeable for $q = 7$, and essentially absent for $q \geq 257$.

C. Effect of repair bandwidth

In both traditional regenerating codes and locally repairable codes the repair bandwidth $d\beta$ is one of the key parameters in the coding rate expression [2], [6]. As mentioned in Section III, for $\beta = \alpha R^RC, R^LRC$ are maximized by $d/n, d/(d + 1)$, respectively. For the minimum-bandwidth regeneration case $\beta = \alpha/d$, the corresponding maximal values are $(d+1)/2n$ (eq. (3)) and $1/2$ [6], respectively. For a test scenario with $n = 12, \alpha = 4, d = 4$ these maximal rates along with the simulated DRNC effective rates ($\text{rank } W^{2000}$) are shown on Figure 4. Expectedly, lower repair bandwidth results in lower effective rate, but the performance gap between RC and DRNC is substantial for both large and small $\beta$.

D. Effect of number of helpers

More helper nodes expectedly slow down the rank decrease, and increases the effective rate, as shown on Figure 5.
When the system operates in volatile conditions, it can perform opportunistic node repair, i.e. download the helper data from all available nodes, so the number of available nodes is random and varies from one repair to another. Figure 5 also shows the rank curves for random number of helpers $d$ with mean $E[d] = 9$, still assuming the uniform node choice. Higher variance of $d$ results in worse effective rates, deterministic $d$ being the best. Increasing variance still performs better than lowering $E[d]$.

E. Effect of localized helpers

In this subsection we consider a generalization of the proposed model, where the storage nodes are grouped into $n/D$ node families, $D$ nodes each, and the helper nodes for a node repair are picked uniformly at random from the local family of the failed node. This generalization covers a spectrum of cases between the original DRNC described in Section II with $D = n$, and LRC with fixed sets of helpers and $D = d + 1$.

Figure 6 shows the simulation results for $n = 24$ nodes and $d = 5$ helpers. Somewhat surprisingly, the effective rate is not monotonous against family size $D$. As the family size goes from $D = 24 = n$ down to $12 \gg d$, diversity of the helper data is decreased, and so is the effective rate. However, as the family size goes down further to $8 \gtrsim d$, the rate goes up again because of the significantly reduced randomness in the helper selection. Finally, when $D$ goes down to $d + 1$, the helper selection randomness vanishes, resulting in the fixed repair sets, the locally repairable regime (LRC), and the highest possible rate $R_{LRC}^{E} = d/(d + 1)$.

V. Conclusions

We explored the simulated performance of DRNC and compared it against the traditional regenerating codes. DRNC provide a significant gain in the effective coding rate with moderate field sizes, and robust performance with random number of the helper nodes. DRNC are also almost-MDS in a sense that decoding is possible from a random set of $B/\alpha$ nodes with a high probability with a proper choice of the code parameters. Overall, DRNC seem to be an efficient storage and repair scheme in volatile networks.

VI. Acknowledgments

This material is based upon work supported by the Air Force Office of Scientific Research (AFOSR) under award No FA9550-14-1-043.

REFERENCES