21st International Symposium on
Mathematical Theory of Networks and Systems
July 7-11, 2014. Groningen, The Netherlands

Erasure codes with simplex locality

Margreta Kuijper[1] and Diego Napp[2]

***Abstract*— We focus on erasure codes for distributed storage. The distributed storage setting imposes locality requirements because of easy repair demands on the decoder. We first establish the characterization of various locality properties in terms of the generator matrix of the code. These lead to bounds on locality and notions of optimality. We then examine the locality properties of a family of nonbinary codes with simplex structure. We investigate their optimality and design several easy repair decoding methods. In particular, we show that any correctable erasure pattern can be solved by easy repair.**

Keywords— simplex codes; erasure decoding; distributed storage coding; locality.

AMS Subject Classifications—94B05; 94B15; 94B35.

I. Introduction

Several classical coding techniques (such as Reed-Solomon erasure codes) are extensively used for data storage, most successfully applied to storage in RAID systems and magnetic recording (see [2]). However, due to the fast-growing demand for large-scale data storage, it would be impossible or extremely expensive to build single pieces of hardware with enough storage capabilities to store the enormous volume of data that is being generated. Hence, new classes of storage technology have emerged using the idea of distributing data across multiple nodes which are interconnected over a network, as we witness in some peer-to-peer (P2P) storage systems [3] and data centers [4] that comprise the backbone infrastructure of cloud computing. We call such systems Networked Distributed Storage Systems (NDSS).

A fundamental issue that arises in this context is the so-called *Repair Problem*: how to maintain the encoded data when failures (node erasures) occur. When a storage node fails, information that was stored in the node is no longer accessible. As a remedy a node is then added to the system to replace the failed node. The added node downloads data from a set of appropriate and accessible nodes to recover the information stored in the failed node. This is called *node repair*. To assess the performance of this repair process, there are several metrics that can be considered: *storage cost*, measured as the amount of data stored in the node, *repair bandwidth*, measured as the total number of bits communicated in the network for each repair and *locality*, measured as the number of nodes needed for each repair. For instance, (n, k) maximum distance separable (MDS) codes are optimal in terms of storage cost since any k nodes contain the minimum amount of information required to recover the original data. However, to repair one single node it is necessary to retrieve information from all k nodes. More specifically, repair is achieved by re-encoding the information from these k nodes and storing part of the re-encoded data in the new node. This results in a poor performance with respect to repair bandwidth as well as locality.

Currently the most well-understood metrics are the repair bandwidth metric and the storage cost metric, see for example [5], [6]. Using network coding techniques, several code constructions have been presented that show optimality with respect to repair bandwidth and storage cost, see [7] and references therein. In contrast, locality is an important metric that has received less attention in the literature. This metric was studied independently by several authors, see [10], [12] and [8] among others, and it is considered to be one of the main repair performance bottlenecks in many NDSS, e.g., in cloud storage applications.

Definition 1.1: An (n, k) code has *locality* r if every codeword symbol in a codeword is a linear combination of at most r other symbols in the codeword.

Thus, when a code of locality r is used then one needs to contact at most r nodes to repair one node. In the recent paper [12] (see also [15]) it was shown that there exists a natural trade-off among redundancy, locality and code minimum distance:

Theorem 1.1: Let C be an (n, k) linear code with minimum distance d and locality r. Then

$$n - k + 1 - d \geq \left\lfloor \frac{k-1}{r} \right\rfloor.$$

Proof: (from [15]; see also [12]) Let G be the generator matrix of C. Choose any $\left\lfloor \frac{k-1}{r} \right\rfloor$ nodes of the code—call these the "leaders". Each leader can be written as a linear combination of at most r other nodes—call this set the "set of friends of the leader". Now define N as the set of nodes which is the union of all sets of friends of the leaders but without the leaders themselves. Then clearly N has less than k elements so that the set of columns in G that corresponds to N spans a space of rank $< k$. Since G has full rank it is possible to enlarge N to a set N' of $\geq k - 1$ columns such that the rank of its corresponding columns equals exactly $k - 1$. Note that because the code has locality r, this enlargement operation can be done without involving any of the leaders. Now define U as the union of N' and the set of leaders. Then

*D. Napp's research has been supported by the Spanish grant DPI2012-31509 and by Portuguese funds through the CIDMA - Center for Research and Development in Mathematics and Applications, and the Portuguese Foundation for Science and Technology ("FCT–Fundação para a Ciência e a Tecnologia"), within project PEst-OE/MAT/UI4106/2014.

[1]Margreta Kuijper is with Faculty of Electrical and Electronic Engineering of the University of Melbourne, Australia mkuijper@unimelb.edu.au

[2]Diego Napp is with the Department of Mathematics, University of Aveiro, Portugal diego@ua.pt

U has at least $k-1+\left\lfloor \frac{k-1}{r} \right\rfloor$ nodes but still, because the code has locality r, the corresponding columns in G span a space of dimension $< k$. By definition of the minimum distance, all $(k\times\cdot)$-submatrices of G that have rank $< k$ must have $\leq n-d$ columns. It therefore follows that $k-1+\left\lfloor \frac{k-1}{r} \right\rfloor \leq n-d$ which proves the theorem. ∎

From the above bound it is seen that MDS codes do not perform well with respect to locality. Indeed, since $d = n-k+1$ they have only trivial locality $r = k$. In contrast, non-MDS codes such as Pyramid codes and Hierarchical codes have been shown to be optimal with respect to the above bound. For these codes the gap $n-k+1-d$ is nonzero due to the fact that they are not MDS. In a sense these codes optimally "use" this gap for locality purposes. More generally, a new class of codes, called locally repairable codes (LRC) [8], [9], [10], [13], addresses the repair problem focusing on minimizing the number of nodes contacted during the repair process.

However, the issue of locality for *multiple* node erasures is less well researched. More specifically, the main problem with the existing locally repairable codes is that although they minimize the number of contacted nodes for the case that only one node has failed, they suffer from the drawback that it is not known how many nodes are needed when several failures occur. This can be due to, for instance, the fact that in these constructions only a single subset of nodes can repair a particular piece of redundant data and therefore if a node from this repair subset is also not available, data cannot be repaired locally, increasing the cost of the repair. Hence, it is desirable to obtain codes providing multiple repair alternatives. Some interesting preliminary results on this problem have been recently presented in [14], seeking to extend the ideas in [12], and in [16] using partial geometry. It is also worth mentioning the results in [10], [11] where some code schemes, akin to the one presented here, are introduced. Below, we indicate by "erasure correcting capability" the classical notion of maximum number of erasures that are guaranteed to be recoverable for the code.

Definition 1.2: Let δ be a positive integer and C be an (n,k) code with erasure correcting capability $\leq \delta$. Then C has δ-*locality* r if, for any erasure pattern with $\leq \delta$ erased symbols, every erased codeword symbol is a linear combination of at most r live symbols in the codeword.

In this paper, we present the simplex code as a code which is especially suitable when locality is relevant and multiple failures may occur. We show that for this code it is always possible to repair each node by simply taking the sum of two live nodes, even in the presence of multiple erasures. To our knowledge this result has not been reported in the literature before and is a useful first step towards more advanced higher rate coding schemes. Our approach is more straightforward than [10], [11] where algebraic Gabidulin type codes are used.

II. Preliminaries

Let $q = 2^m$ and let $\mathbb{F}_q$ be a finite field with q elements. A q-ary linear (n,k)-code $\mathcal{C}$ of length n and rank k is a k-dimensional linear subspace of $\mathbb{F}_q^n$. Full-rank matrices $G \in \mathbb{F}_q^{k\times n}$ and $H \in \mathbb{F}_q^{n\times(n-k)}$ with the property that

$$\begin{aligned} \mathcal{C} &= \text{Im}_{\mathbb{F}_q} G = \left\{c = uG \in \mathbb{F}_q^n : u \in \mathbb{F}_q^k\right\} \\ &= \ker_{\mathbb{F}_q} H = \left\{c \in \mathbb{F}_q^n : Hc^\top = 0\right\}, \end{aligned}$$

are called *generator matrix* and *parity-check matrix*, respectively.

Definition 2.1: Let k be a positive integer, $n = 2^k - 1$ and let G be a $k \times n$ matrix whose columns are the distinct non-zero vectors of $\mathbb{F}_2^k$. Let C be the binary code over $\mathbb{F}_2$ that has G as its generator matrix. Then C is called a binary *simplex* (n,k)-code.

Binary simplex codes are classical codes with minimum distance $d = 2^{k-1}$ and they are dual to the binary Hamming codes.

Example 2.1: Let $k = 3$ and $\mathcal{C} \subset \mathbb{F}_2^7$ be a simplex $(7,3)$-code. Then, its generator matrix G and parity-check matrix H are given by;

$$G = \begin{pmatrix} 1 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \end{pmatrix}$$

and

$$H = \begin{pmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 \end{pmatrix},$$

respectively.

When coding is used in distributed storage systems, a data object or file $u = (u_1, u_2, \ldots, u_k) \in \mathbb{F}_q^k$ of k symbols is redundantly stored across n different nodes in $c = (c_1, c_2, \ldots, c_n) = uG \in \mathbb{F}_q^n$, where G is the generator matrix of an (n,k)-code $\mathcal{C}$.

Let $S = \{c_1, \ldots, c_n\}$ be the set of nodes, $S^e \subset S$ the set of erased notes, $S^\ell \subset S$ the set of live nodes and $S_i^e = \{i \mid c_i \in S^e\}$ and $S_i^\ell = \{i \mid c_i \in S^\ell\}$ the indices of the erased and live nodes, respectively. A node c_i is said to be *related* to the pair (c_j, c_k) if $c_i = c_j + c_k$, $i,j,k \in \{1,\ldots,n\}$. Two pairs of nodes are said to be *disjoint* if they do not share a common node. If $c_i \in S^e$ is related to the pair (c_j, c_k) where $c_j, c_k \in S^\ell$, then it is said that c_i *allows for easy repair*.

In terms of computational complexity, this implies that the cost of a node reconstruction is that of a simple addition of two nodes.

Note that if c_i is related to the pair (c_j, c_k), then c_j is related to (c_i, c_k) and c_k is related to (c_j, c_i).

The following lemma is useful for the sequel of the paper.

Lemma 2.1: Let C be a linear (n,k) code and let S_i^e denote the set of indices of erased nodes. Denote $|S_i^e| = n - s$. Then the following statements are equivalent:

1) The erasure pattern corresponding to S_i^e is correctable;
2) The $k\times s$ matrix $\hat{G}$ formed by deleting the i-th columns of G where $i \in S_i^e$, is right invertible;

3) The $(n-k)\times(n-s)$ matrix $\hat{H}$ formed by the i-th columns of H where $i \in S_i^e$, is left invertible.

Proof: Denote the set of indices of the live (=non-erased) nodes by S_i^ℓ. Clearly 1) holds if and only if there do not exist two different codewords that coincide in positions corresponding to S_i^ℓ. Since the code is linear this is equivalent to the non-existence of a nonzero codeword whose symbols at positions in S_i^ℓ are zero. The latter is clearly equivalent to the linear independence of the columns of $\hat{H}$. Next, we prove the equivalence of 1) and 2). Write $S_i^\ell = \{j_1, \ldots, j_s\}$. Consider the system of equations

$$\left[\begin{array}{ccc} c_{j_1} & \cdots & c_{j_s} \end{array}\right] = u\hat{G}.$$

The solvability of this system is equivalent to the recovery of u and the repair of all erasures. The equivalence of 1) and 2) now follows from the fact that this system is solvable for any erasure pattern that corresponds to S_i^e if and only if $\hat{G}$ is right invertible. ∎

III. Simplex locality

In this section we propose a nonbinary simplex code, defined as follows:

Let $G \in \mathbb{F}_2^{k\times n}$ and $H \in \mathbb{F}_2^{n\times(n-k)}$ be the generator matrix and parity-check matrix of a binary simplex (n,k)-code over $\mathbb{F}_2$. Via this generator matrix G we encode the data to be stored $u \in \mathbb{F}_q^k$ to

$$c = (c_1, c_2, \ldots, c_n) = uG \in \mathbb{F}_q^n, \tag{1}$$

with $q = 2^m$ for some $m \in \mathbb{N}$. The resulting code is an (n,k) code over F_q that we call a *simplex code over* F_q.

It is easy to see that these codes inherit their distance property from the binary simplex codes, namely $d = 2^{k-1}$. The codes also possess several good locality properties, starting with the next lemma which is based on a wellknown property of the binary simplex code.

Lemma 3.1: Let C be an (n,k) simplex code over F_q. Denote its set of nodes by $S = \{c_1, \ldots, c_n\}$. Then, each node $c_i \in S$, $i \in \{1, \ldots, n\}$ is related to $\frac{n-1}{2}$ different pairs.

Proof: Choose any node, say $\hat{c} \in S$, and let $\hat{S} := \{\hat{c}\}$. Take $c_{1_j} \in S \setminus \hat{S}$ and set $c_{1_k} = c_{1_j} + \hat{c}$. Due to the fact that any sum of two columns of G is another column of G we have that for any $c_j, c_k \in S$, $c_j + c_k \in S$. Hence, $c_{1_k} \in S$ and let $\hat{S} := \hat{S} \bigcup \{c_{1_j}, c_{1_k}\}$. Again take any node $c_{2_j} \in S \setminus \hat{S}$ and set $c_{2_k} = c_{2_j} + \hat{c} \in S$. Note that $c_{2_k} \notin \hat{S}$ and therefore the pairs (c_{1_j}, c_{1_k}) and (c_{2_j}, c_{2_k}) are disjoint. Let $\hat{S} := \hat{S} \bigcup \{c_{2_j}, c_{2_k}\}$ and the cardinality of the set $\hat{S}$ is increased by two in each step. Repeating this process $\frac{n-1}{2}$ times, we obtain $\frac{n-1}{2}$ disjoint pairs related to $\hat{c}$. Since the choice of $\hat{c}$ is arbitrary, this concludes the proof. ∎

It follows from the above lemma that a simplex code over F_q has locality 2. Therefore any single erasure pattern allows for easy repair. The next theorem, reminiscent of Corollary 3 in [10], shows that this is also true for multiple erasure patterns that are within the code's erasure correcting capability.

Theorem 3.1: Let C be an (n,k) simplex code over F_q. Denote the set of erased nodes by S^e. If $|S^e| \leq \frac{n-1}{2}$, then *all* the nodes in S^e allow for easy repair. Thus, the $\frac{n-1}{2}$-locality of C equals 2.

Proof: If $|S^e| \leq \frac{n-1}{2}$ then we have $n - |S^e| > \frac{n-1}{2}$ live nodes. By Lemma 3.1 any erased node is related to $\frac{n-1}{2}$ disjoint pairs which implies that at least one of these pairs is comprised of two live nodes, which implies easy repair. ∎

Note that the above theorem deals with erasure patterns whose number of erasures are within the erasure correcting capability of the code. We now turn our attention to the larger class of erasure patterns that are correctable and possibly have $> \frac{n-1}{2}$ erasures.

Lemma 3.2: Let C be an (n,k) simplex code over F_q. Denote the set of erased nodes by S^e. Then if S^e corresponds to a correctable erasure pattern then there exists an erased node that allows for easy repair.

Proof: Denote again the set of live nodes by S^ℓ; denote the set of its indices by $S_i^\ell = \{j_1, \ldots, j_s\}$. Consider the system of equations

$$\left[\begin{array}{ccc} c_{j_1} & \cdots & c_{j_s} \end{array}\right] = u\hat{G},$$

where $\hat{G}$ is the $k \times s$ matrix that remains after deleting from G the i-th columns at erased positions. Since S^e corresponds to a correctable erasure pattern, this system of equations is solvable over $\mathbb{F}_q$. Thus it follows from Lemma 2.1 that $\hat{G}$ has rank k. As a result, for all $\hat{c} \in S^e$ there exist an integer g and $a_j \in S_i^\ell$ for $j = 1, \ldots, g$ such that

$$\hat{c} = c_{a_1} + \cdots + c_{a_g}.$$

If $c_{a_1} + c_{a_2} \in S^e$, then we have found one erased node that is easily reparable. If not, *i.e.* if $c_{a_1} + c_{a_2} \in S^\ell$, then denote $c_{b_1} = c_{a_1} + c_{a_2}$ and therefore $\hat{c} = c_{b_1} + c_{a_3} + \cdots + c_{a_g}$. Again, if $c_{b_1} + c_{a_3} \in S^e$, then we have found one erased node that is easily reparable. If not, *i.e.*, if $c_{b_1} + c_{a_3} \in S^\ell$, then denote $c_{b_2} = c_{b_1} + c_{a_3}$ and therefore $\hat{c} = c_{b_2} + c_{a_4} + \cdots + c_{a_g}$. This process must end yielding either that $c_{b_j} + c_{a_{j+2}} \in S^e$ with $c_{b_j}, c_{a_{j+2}} \in S^\ell$ for some $j \in \{1, 2, \ldots, g-3\}$ or $\hat{c} = c_{b_{g-2}} + c_{a_g}$. In both cases we obtain an easy repair and the proof is completed. ∎

In the following algorithm we present an "easy repair of one node" algorithm for an encoded file $c = (c_1, \ldots, c_n)$ with node failures. The generator matrix used for the codification is $G = \left[\begin{array}{ccc} G_1 & \cdots & G_n \end{array}\right]$ where its columns G_i are elements of $\mathbb{F}_{2^k}$.

Above, $findlivenodes(c)$ is a function that returns a vector $(j_1, \ldots, j_s)$ of live nodes indices. The algorithm returns two live nodes c_{j_i} and c_{j_t} that repair an erased node $c_{j_i} + c_{j_t}$ and the Boolean variable "Correctable" that takes the values TRUE or FALSE; in case the erasure pattern is uncorrectable, the algorithm returns zero values and the statement "FALSE"

Data: $(c_1, \ldots, c_n)$ and $G = \begin{bmatrix} G_1 & \cdots & G_n \end{bmatrix}$.
Result: $(c_{j_i}, c_{j_t}, c_{j_i} + c_{j_t}, \text{Correctable})$.
$(j_1, \ldots, j_s) = findlivenodes(c)$;
$s = length(findlivenodes(c))$;
$G^\ell = \{G_{j_1}, \ldots, G_{j_s}\}$;
$i = 1,\ t = 2$;
while $[G_{j_i} + G_{j_t} \in G^\ell]$ *AND* $[i < s - 1]$ **do**
  if $t = s$ **then**
    $t = i + 2,\ i = i + 1$
  else
    $t = t + 1$
  end
end
if $[i = s - 1]$ *AND* $[G_{j_i} + G_{j_t} \in G^\ell]$ **then**
  $(c_{j_i}, c_{j_t}, c_{j_i} + c_{j_t}$, Correctable=TRUE)
else
  $(c_{j_{i-1}} = 0, c_{j_i} = 0, c_{j_{i-1}} + c_{j_i} = 0,$
  Correctable=FALSE)
end

Theorem 3.2: Let C be an (n, k) simplex code over F_q with S^e denoting the set of erased nodes. If S^e corresponds to a correctable erasure pattern, then repeated application of Algorithm 1 recovers all erasures by easy repairs.

Proof: Because of Lemma 3.2, an easy repair situation exists. The algorithm is clearly defined in such a way that it will find this easy repair situation and carry out the repair. Once repaired, the erasure pattern is of course still correctable and we repeat over and over until all erasures are recovered. ∎

Example 3.1: Consider the matrices G and H as defined in Example 2.1. Suppose that we have a file $u \in \mathbb{F}_q^3$ to be stored in 7 nodes, *i.e.*, $uG = c = (c_1, \ldots, c_7) \in \mathbb{F}_q^7$, and that erasures occur in nodes c_1, c_2, c_4 and c_6, *i.e.*, $S_i^e = \{1, 2, 4, 6\}$, $S_i^\ell = \{3, 5, 7\}$. Thus the pattern is correctable despite the fact that the number of erasures is outside of the erasure correcting capability. It now follows from Lemma 3.2 that there exists an erased node that allows for easy repair. Indeed, $c_1 = c_3 + c_5$ and in fact there exist several erased nodes that allow for easy repair, namely also $c_2 = c_5 + c_7$ and $c_4 = c_3 + c_7$. By Lemma 3.2, we can repair all nodes by easy repairs. Indeed, c_2 and c_4 already allow for easy repair and, once c_2 is repaired, we repair c_6 from $c_6 = c_2 + c_3$.

Remark 3.1: Note that in the previous example the node c_6 cannot be the first node to be easily repaired and we need to repair a different node first. However, when the number of erasures does not exceed the erasure correcting capability of the code, then *any* erased node can be chosen to start the repair, thus allowing for parallelization of easy repairs.

IV. Conclusions

In this paper, we have presented the family of non-binary simplex codes as a family which is highly suitable for efficient erasure coding in multiple-erasure settings within Networked Distributed Storage Systems. These non-binary codes are constructed using the generator matrix of binary simplex codes and hence inherit excellent locality properties even in the presence of large erasure patterns. They allow for easy encoding, requiring only addition operations in F_q. We have shown that if the erasure pattern is correctable at all, then it is possible to repair each of the failed nodes by simply taking the sum of two live nodes. A drawback of the simplex code is its low rate, particularly for high value of n. Further design and analysis of higher rate coding schemes based on simplex codes is part of our ongoing and future work.

References

[1] H. Gluesing-Luerssen, J. Rosenthal, and R. Smarandache, Strongly MDS convolutional codes. IEEE Trans. Inform. Theory, 52(2):584–598, 2006.
[2] M. Blaum, J. Brady, J. Bruck, and J. Menon, EVENODD: An efficient scheme for tolerating double disk failures in RAID architectures, IEEE Trans. Comput., vol. C-44, no. 2, pp. 192202, Feb. 1995.
[3] www.wuala.com
[4] C. Huang, H. Simitci, Y. Xu, A. Ogus, B. Calder, P. Gopalan,J. Lin, S. Yekhanin, Erasure Coding in Windows Azure Storage.
[5] A. G. Dimakis, P. B. Godfrey, Y. Wu, M. Wainwright and K. Ramchandran, Network Coding for Distributed Storage Systems IEEE Transactions on Information Theory, Vol. 56, Issue 9, Sept. 2010.
[6] B. Gaston , P. Pujol, M. Villanueva, Quasi-cyclic regenerating codes IEEE Transactions on Information Theory (under revision), arXiv:1209.3977[cs.IT], 2013.
[7] A. Dimakis, K. Ramchandran, and Y. Wu, A survey on network codes for distributed storage, IEEE Trans. Inf. Theory, vol. 99, pp. 12041216, 2011.
[8] D. Papailiopoulos and A. Dimakis, Locally repairable codes, in Proc. of the IEEE ISIT, 2012.
[9] D. S. Papailiopoulos and A. G. Dimakis, Storage codes with optimal repair locality, in Proceedings of the IEEE Intl. Symposium on Information Theory (ISIT), 2012.
[10] F. Oggier, A. Datta, Self-repairing Homomorphic Codes for Distributed Storage Systems",The 30th IEEE International Conference on Computer Communications, INFOCOM 2011. Extended version at http://arxiv.org/abs/1107.3129.
[11] F. Oggier, A. Datta, Self-Repairing Codes for Distributed Storage - A Projective Geometric Construction, In IEEE Information Theory Workshop (ITW) 2011.
[12] P. Gopalan, C. Huang, H. Simitci, and S. Yekhanin, On the locality of codeword symbols, IEEE Transactions on Information Theory, vol. 58, pp. 69256934, Nov 2012.
[13] G. Kamath, N. Prakash, V. Lalitha, and P. Kumar, Codes with local regeneration, arXiv preprint arXiv:1211.1932, 2012.
[14] N. Prakash, G. M. Kamath, V. Lalitha, and P. V. Kumar, Optimal linear codes with a local-error-correction property, in Proc. IEEE Int. Symp. Inf. Theory (ISIT), Cambridge, MA, Jul. 2012, pp. 27762780.
[15] I. Tamo and A. Barg, A family of optimal locally recoverable codes, arXiv preprint arXiv:1311.3284, 2013.
[16] L. Pamies-Juarez, H.D.L. Hollmann, F. Oggier, "Locally Repairable Codes with Multiple Repair Alternatives,", IEEE International Symposium on Information Theory (ISIT 2013).