

# On Cooperative Local Repair in Distributed Storage

Ankit Singh Rawat  
Department of ECE  
The University of Texas at Austin  
Austin, TX 78712  
Email: ankitsr@utexas.edu.

Arya Mazumdar  
Department of ECE  
University of Minnesota – Twin Cities  
Minneapolis, MN 55455  
Email: arya@umn.edu.

Sriram Vishwanath  
Department of ECE  
The University of Texas at Austin  
Austin, TX 78712  
Email: sriram@utexas.edu.

**Abstract**—Erasure-correcting codes, that support *local repair* of codeword symbols, have attracted substantial attention recently for their application in distributed storage systems. In this paper we study a generalization of the usual locally recoverable codes. We consider such codes that any small set of codeword symbols is recoverable from a small number of other symbols. We call this *cooperative local repair*. We present bounds on the dimension of such codes as well as give explicit constructions of families of codes. Some other results regarding cooperative local repair are also presented, including an analysis for the Hadamard codes.

## I. INTRODUCTION

In this paper we explore a new class of codes that enable efficient recovery from the failure (erasure) of a small number of code symbols. In particular, we study codes with  $(r, \ell)$ -locality which allow for any  $r$  failed code symbols to be recovered by contacting  $\ell$  other intact code symbols. Our study of such codes is motivated by their application in distributed storage systems *a.k.a.* cloud storage, where information is stored over a network of storage nodes (disks). In distributed storage systems, one has to introduce redundancy in order to protect the stored information against inevitable node (disk) failures.

The traditional erasure codes are highly sub-optimal for distributed storage setting as these codes incur high cost of code repair [1]. Recently, multiple classes of erasure codes have been proposed to optimize the performance of code repair process with respect to various metrics. In particular, the codes that minimize *repair-bandwidth*, *i.e.*, the number of bits communicated during a node repair, are studied in [1]–[4] and references therein. Another family of erasure codes that focus on small *locality*, *i.e.*, enabling repair of single code symbol by contacting a small number of other code symbols, are presented in [5]–[9].

A code is said to have *all-symbol locality* of  $r$  if every code symbol is a function of at most  $r$  other code symbols. This ensures *local repair* of each code symbol by contacting at most  $r$  other code symbols. In this paper we generalize the notion of codes with all-symbol locality to codes with  $(r, \ell)$ -locality: any set of  $\ell$  code symbols are functions of at most  $r$  other code symbols. This allows for *cooperative local repair* of code symbols, where a groups of  $\ell$  code symbols is repaired by contacting at most  $r$  other code symbols. The approach of cooperative code repair has been previously explored in the

context of repair-bandwidth efficient codes in [10], [11] and references therein.

In this paper we address two important issues regarding codes with  $(r, \ell)$ -locality: 1) obtaining trade-offs among minimum distance, dimension, and locality parameters  $(r, \ell)$  for such code; and 2) presenting explicit constructions for codes with  $(r, \ell)$ -locality that are close to the obtained trade-offs. We present a formal definition of codes with  $(r, \ell)$ -locality in Section II. In Section III, we obtain an upper bound on the minimum distance of a code with  $(r, \ell)$ -locality which encodes  $k$  information symbols to  $n$  symbols long codewords. We then bound the best possible rate for a code with  $(r, \ell)$ -locality. In Section IV we present two constructions for the codes that have  $(r, \ell)$ -locality and comment on their rates with respect to the bound obtained in Section III. In Section IV-D, we study the punctured Hadamard codes (*a.k.a.* Simplex codes) in the context of cooperative local repair. In Section IV-E we generalize the Partition code framework from Section IV-A. We briefly consider the setting of random erasures in Section IV-F.

A short note on notation: we use bold lower case letters to denote vectors. For an integer  $n \geq 1$ ,  $[n]$  denotes the set  $\{1, 2, \dots, n\}$ .

## II. CODES WITH $(r, \ell)$ -LOCALITY

**Definition 1.** Let  $\mathcal{C}$  be an  $(n, k)$  code. We call  $\mathcal{C}$  to be an  $(n, k)$  code with  $(r, \ell)$ -locality if for each  $\mathcal{S} \subset [n]$  with  $|\mathcal{S}| = \ell$ , we have a set  $\Gamma_{\mathcal{S}} \subseteq [n] \setminus \mathcal{S}$  such that

- 1)  $|\Gamma_{\mathcal{S}}| \leq r$ ,
- 2) For any codeword  $\mathbf{c} = (c_1, c_2, \dots, c_n) \in \mathcal{C}$ , the  $\ell$  code symbols  $\mathbf{c}_{\mathcal{S}} := \{c_i : i \in \mathcal{S}\}$  are functions of the code symbols  $\mathbf{c}_{\Gamma_{\mathcal{S}}} := \{c_i : i \in \Gamma_{\mathcal{S}}\}$ .

Note that Definition 1 ensures that any  $\ell$  code symbols can be *cooperatively* repaired from at most  $r$  other code symbols. This generalizes the notion of codes with *all-symbol locality*  $r$  [5], [6], [12], [13], where locality is defined with respect to one code symbol, *i.e.*,  $\ell = 1$ .

**Remark 1.** For a code  $\mathcal{C}$  with all-symbol locality  $r$ , we have the following bound on its minimum distance [5], [6].

$$d_{\min}(\mathcal{C}) \leq n - k - \left\lceil \frac{k}{r} \right\rceil + 2. \quad (1)$$

The code attaining the bound in (1) are presented in [6]–[9] and references therein.

### III. RATE VS. DISTANCE TRADE-OFF FOR CODES WITH $(r, \ell)$ -LOCALITY

In this section, for given  $r$  and  $\ell$ , we present a trade-off between the rate and the minimum distance of a code with  $(r, \ell)$ -locality (cf. Definition 1). We employ the general proof technique introduced in [5], [14], [15] to obtain the following result.

**Theorem 1.** *Let  $\mathcal{C}$  be an  $(n, k)$  code (linear, or non-linear) with  $(r, \ell)$ -locality. Then, the minimum distance of  $\mathcal{C}$  satisfies*

$$d_{\min}(\mathcal{C}) \leq n - k + 1 - \ell \left( \left\lceil \frac{k}{r} \right\rceil - 1 \right). \quad (2)$$

---

**Algorithm:** Construction of sub-code  $\mathcal{C}' \subset \mathcal{C}$ .

---

**Input:**  $(n, k)$  code  $\mathcal{C} \subseteq \mathbb{F}_q^n$  with  $(r, \ell)$ -locality.

```

1:  $\mathcal{C}_0 = \mathcal{C}$ 
2:  $j = 0$ 
3: while  $|\mathcal{C}_j| > q^\ell$  do
4:    $j = j + 1$ .
5:   Choose  $i_1^j, i_2^j, \dots, i_\ell^j \in [n]$  such that, for every  $m \in [\ell]$ , there exist at least two codewords in  $\mathcal{C}_{j-1}$  that differ at  $i_m^j$ -th coordinate.
6:   Let  $\mathcal{R}_j = \Gamma_{\{i_1^j, \dots, i_\ell^j\}}$  be the index of at most  $r$  code symbols that cooperatively repair  $\ell$  code symbols indexed by  $\{i_1^j, \dots, i_\ell^j\}$ .
7:   Let  $\mathbf{y} \in \mathbb{F}_q^{|\mathcal{R}_j|}$  be the most frequent element in the multi-set  $\{\mathbf{x}_{\mathcal{R}_j} : \mathbf{x} \in \mathcal{C}_{j-1} \subset \mathbb{F}_q^n\}$ .
8:   Define  $\mathcal{C}_j := \{\mathbf{x} : \mathbf{x} \in \mathcal{C}_{j-1} \subset \mathbb{F}_q^n \text{ and } \mathbf{x}_{\mathcal{R}_j} = \mathbf{y}\}$ .
9:   if  $1 < |\mathcal{C}_j| \leq q$  then
10:      $\mathcal{C}' = \mathcal{C}_j$ .
11:   end while
12:   else if  $|\mathcal{C}_j| = 1$  then
13:     Pick a maximal subset  $\tilde{\mathcal{R}}_j \subseteq \mathcal{R}_j$  such that  $|\tilde{\mathcal{C}}_j| > 1$ , where  $\tilde{\mathcal{C}}_j := \{\mathbf{x} : \mathbf{x} \in \mathcal{C}_{j-1} \subset \mathbb{F}_q^n, \mathbf{x}_{\tilde{\mathcal{R}}_j} = \tilde{\mathbf{y}}_j\}$  and  $\tilde{\mathbf{y}}_j \in \mathbb{F}_q^{|\tilde{\mathcal{R}}_j|}$  be the most frequent element in the multi-set  $\{\mathbf{x}_{\tilde{\mathcal{R}}_j} : \mathbf{x} \in \mathcal{C}_{j-1} \subset \mathbb{F}_q^n\}$ .
14:      $\mathcal{C}' = \tilde{\mathcal{C}}_j$ .
15:   end while.
16: end if
17: end while

```

**Output:**  $\mathcal{C}'$ .

---

Fig. 1: Construction of sub-code  $\mathcal{C}' \subset \mathcal{C}$ .

*Proof:* The proof involves construction of a subcode  $\mathcal{C}' \subset \mathcal{C}$  such that all but a small number of coordinates in every codeword of  $\mathcal{C}'$  are fixed. Note that we have

$$d_{\min}(\mathcal{C}) \leq d_{\min}(\mathcal{C}'). \quad (3)$$

Given  $\mathcal{C}'$ , one can obtain a code  $\mathcal{C}''$  with  $|\mathcal{C}''| = |\mathcal{C}'|$  by removing fixed coordinates from all the codeword in  $\mathcal{C}'$ . This

implies that  $d_{\min}(\mathcal{C}'') = d_{\min}(\mathcal{C}')$ , which along with (3) give us the following.

$$d_{\min}(\mathcal{C}) \leq d_{\min}(\mathcal{C}''). \quad (4)$$

We describe the construction of the subcode  $\mathcal{C}'$  in Fig. 1. Before proceeding with the analysis, we argue the correctness of the algorithm in Fig. 1. Note that it is always possible to find  $\ell$  coordinates  $\{i_1^j, i_2^j, \dots, i_\ell^j\}$  at line 5. When the algorithm reaches line 5, the subcode  $\mathcal{C}_{j-1}$  has more than  $q^\ell$  codewords. Therefore, there must be at least  $\ell$  coordinates in the codewords in  $\mathcal{C}_{j-1}$  that are not fixed in the previous iterations. This also implies that, for  $m \in [\ell]$ ,

$$i_m^j \notin \mathcal{I}_{j-1} := \bigcup_{j' \in [j-1]} \left( \mathcal{R}_{j'} \cup \{i_1^{j'}, \dots, i_\ell^{j'}\} \right) \subset [n]. \quad (5)$$

Note that the code symbols indexed by  $\mathcal{I}_{j-1}$  are fixed in  $\mathcal{C}_{j-1}$ . Thus, we must have

$$\mathcal{R}_j = \Gamma_{\{i_1^j, \dots, i_\ell^j\}} \not\subset \mathcal{I}_{j-1}.$$

Otherwise,  $\{i_1^j, \dots, i_\ell^j\}$  would have been fixed in the previous iterations. Assuming that the construction in Fig. 1 ends in  $t$ -th iteration, we obtain in Appendix A that

$$t \geq \left\lceil \frac{k}{r} \right\rceil - 1. \quad (6)$$

and

$$\log_q |\mathcal{C}'| \geq k - |\mathcal{I}_t| + t\ell. \quad (7)$$

Now, we define  $\mathcal{C}'' = \mathcal{C}'|_{\mathcal{I}_t}$  which denotes the code obtained by puncturing the codewords in  $\mathcal{C}'$  at the coordinates associated with the set  $\mathcal{I}_t$ . We have  $|\mathcal{C}''| = |\mathcal{C}'|$  and  $d_{\min}(\mathcal{C}'') = d_{\min}(\mathcal{C}')$ . Moreover, the length of the codewords in  $\mathcal{C}''$  is  $n - |\mathcal{I}_t|$ . Next, applying the Singleton bound on  $\mathcal{C}''$  gives us

$$\begin{aligned} d_{\min}(\mathcal{C}) &\leq d_{\min}(\mathcal{C}'') \leq n - |\mathcal{I}_t| - \log_q |\mathcal{C}''| + 1 \\ &\leq n - |\mathcal{I}_t| - (k - |\mathcal{I}_t| + t\ell) + 1 \\ &= n - k - t\ell + 1, \end{aligned} \quad (8)$$

It follows from (8) and (6) that

$$d_{\min}(\mathcal{C}) \leq n - k + 1 - \ell \left( \left\lceil \frac{k}{r} \right\rceil - 1 \right). \quad (9)$$

The analysis of the minimum distance when the construction in Fig 1 ends at line 15 is presented in Appendix A. This completes the proof.  $\blacksquare$

Note that an  $(n, k)$  code with  $(r, \ell)$ -locality has its minimum distance at least  $\ell + 1$  as it can recover from the erasure of any  $\ell$  code symbols (cf. Definition 1). Combining this observation with Theorem 1, we obtain the following result.

**Corollary 1.** *The rate of an  $(n, k)$  code with  $(r, \ell)$ -locality is bounded as*

$$\frac{k}{n} \leq \frac{r}{r + \ell}. \quad (10)$$

*Proof:* It follows from (2) and the fact  $d_{\min}(\mathcal{C}) \geq \ell + 1$  that

$$\ell + 1 \leq d_{\min}(\mathcal{C}) \leq n - k + 1 - \ell \left( \left\lceil \frac{k}{r} \right\rceil - 1 \right). \quad (11)$$

Using  $\left\lceil \frac{k}{r} \right\rceil - 1 \geq \frac{k}{r} - 1$ , we get

$$k + \ell \frac{k}{r} \leq n,$$

or

$$\frac{k}{n} \leq \frac{r}{r + \ell}.$$

#### IV. CONSTRUCTION OF CODES WITH $(r, \ell)$ -LOCALITY

In this section we address the issue of constructing high rate codes that have  $(r, \ell)$ -locality. In particular, we describe two simple constructions that ensure cooperative local repair for the failure of any  $\ell$  code symbols: 1) Partition code and 2) Product code. In Partition code, we partition the information symbols in groups of  $\frac{r}{\ell}$  symbol and encode each group with an  $(\frac{r}{\ell} + \ell, \frac{r}{\ell})$ -MDS code (cf. Section IV-A. On the other hand, a product code is obtained by arranging  $k = \left(\frac{r}{\ell}\right)^\ell$  information symbols in an  $\ell$ -dimensional array and then introducing parity symbols along different dimensions of the array (cf. Section IV-B).

##### A. Partition Code

For the ease of exposition, we assume that  $\ell | r$ ,  $\frac{r}{\ell} \leq \ell$ , and  $\left(\frac{r}{\ell}\right) | k$ . Given  $k$  information symbol over  $\mathbb{F}_q$ , a Partition code encodes these symbols into  $n = k \frac{r + \ell^2}{r}$  symbols long codewords as follows (see Fig. 2):

- 1) Partition  $k$  information symbols into  $g = \frac{k\ell}{r}$  groups of size  $\frac{r}{\ell}$  each.
- 2) Encode the symbols in each of the  $g$  groups using an  $(\frac{r}{\ell} + \ell, \frac{r}{\ell})$ -MDS code over  $\mathbb{F}_q$ . We refer to the  $r + \ell$  code symbols obtained by encoding  $r$  information symbol in the  $i$ -th group as  $i$ -th local group.

As it is clear from the construction, Partition code has rate  $\frac{k}{n} = \frac{r}{r + \ell^2}$ . Moreover, a code symbol can be recovered from any  $\frac{r}{\ell}$  other code symbol from its local group. In the worst case, when  $\ell$  failed code symbols belong to  $\ell$  distinct local group, we can recover all  $\ell$  symbols from  $\ell \frac{r}{\ell} = r$  code symbols, downloading  $\frac{r}{\ell}$  symbols from each of the  $\ell$  local groups containing one failed code symbol.

##### B. Product Code

Product codes are a well known family of codes in the coding theory literature. Given  $k = \left(\frac{r}{\ell}\right)^\ell$  information symbols and  $\ell | r$ , we first arrange  $k = \left(\frac{r}{\ell}\right)^\ell$  information symbols in an  $\ell$ -dimensional array with index of each dimension of the array ranging in the set  $\left[\frac{r}{\ell}\right]$ . These information symbols are then encoded to obtain an  $n = \left(\frac{r}{\ell} + 1\right)^\ell$  symbols long code word. In the following we describe the encoding process for  $\ell = 2$ -dimensional array (see Fig. 3). The generalization of the encoding process for higher dimensions is straightforward.

- 1) Arrange  $k = \left(\frac{r}{2}\right)^2$  information symbols in an  $\frac{r}{2} \times \frac{r}{2}$  array.
- 2) For each row of the array, add a parity symbol by summing all  $\frac{r}{2}$  symbols in the row and append these symbols to their respective rows.
- 3) For each of the  $\frac{r}{2} + 1$  columns of the updated array, add a parity by summing all  $\frac{r}{2}$  symbols in the column.

$m_1$	$m_2$	$\dots$	$m_{\frac{r}{2}}$	$p_1^r$
$m_{\frac{r}{2}+1}$	$m_{\frac{r}{2}+2}$	$\dots$	$m_{2\frac{r}{2}}$	$p_2^r$
$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
$m_{\gamma+1}$	$m_{\gamma+2}$	$\dots$	$m_{\frac{r}{4}}$	$p_{\frac{r}{2}}^r$
$p_1^c$	$p_2^c$	$\dots$	$p_{\frac{r}{2}}^c$	$p_{\frac{r}{2}+1}^c$

Fig. 3: Description of Product code construction with  $\ell = 2$ .  $\{p_1^r, p_2^r, \dots, p_{\frac{r}{2}}^r\}$  denote the parity nodes associated with each row of the two dimensional array of information symbols.  $\{p_1^c, p_2^c, \dots, p_{\frac{r}{2}+1}^c\}$  represent the parity symbols for the columns of the updated array. We use  $\gamma$  to denote  $\left(\frac{r}{2} - 1\right) \frac{r}{2}$ .

##### C. Comparison between Partition and Product code

We now compare the rate of Partition code and Product code with the bound in (10). For any  $\ell \geq 1$ , we have

$$\left(\frac{r}{r + \ell}\right)^\ell \leq \frac{r}{r + \ell^2}. \quad (12)$$

Note that (12) follows from the fact that

$$r^\ell (r + \ell^2) \leq r(r^\ell + \ell r^{\ell-1} \ell) + r \left( \sum_{i=2}^{\ell} \binom{\ell}{i} r^{\ell-i} \ell^i \right) = r(r + \ell)^\ell.$$

Therefore, Partition code approach provides  $(r, \ell)$ -locality with a better rate. However, for all system parameters, the rate of Partition code is smaller than the known bound (10), *i.e.*,

$$\frac{r}{r + \ell^2} \leq \frac{r}{r + \ell}.$$

Here, we would like to note that the difference between the rate achieved by the Partition code and the bound in (10) gets smaller as the parameter  $r$  becomes large as compared to the parameter  $\ell$ . It is an interesting problem to either tighten the bound in (10) or present a construction for codes with  $(r, \ell)$ -locality which have higher rate than that of Partition code.

##### D. Cooperative Local Repair for Hadamard Codes

Here, we comment on the local repairability of punctured Hadamard codes. Punctured Hadamard codes are also referred to as Simplex codes, which are the dual codes of Hamming codes. In particular, we show that an  $[n = 2^k - 1, k, 2^{k-1}]_2$  punctured Hadamard code has  $(r = \ell + 1, \ell)$  locality for any  $\ell \leq \frac{n-1}{2}$ . An  $[n = 2^k - 1, k, 2^{k-1}]_2$  punctured Hadamard code encodes a  $k$  bits long message  $(m_1, m_2, \dots, m_k)$  to an  $n = 2^k - 1$  codeword  $\mathbf{c} = (c_1, c_2, \dots, c_{n=2^k-1})$  such that

$$c_i = \sum_{j=1}^k m_j b_j^i \pmod{2}.$$

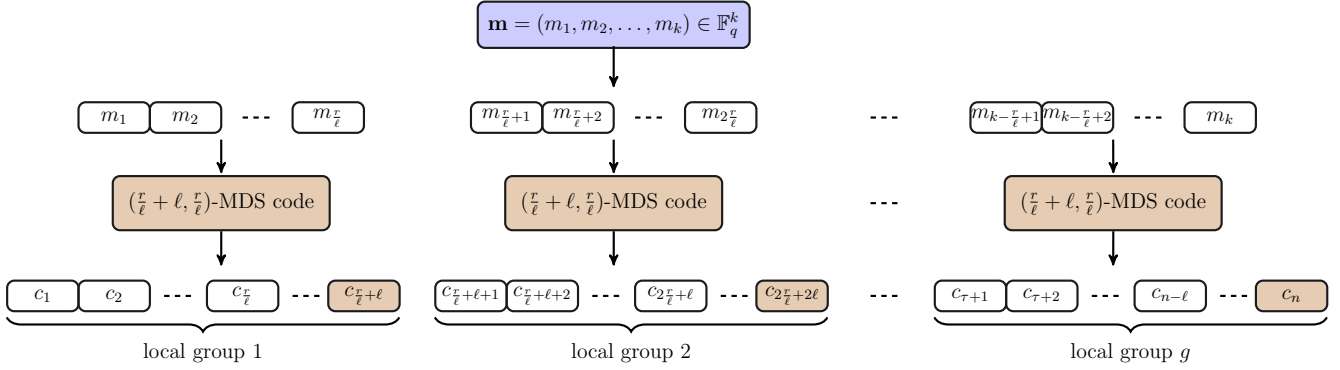


Fig. 2: Illustration of the encoding process of Partition code.

Here  $\mathbf{b}^i = (b_1^i, b_2^i, \dots, b_k^i) \in \mathbb{F}_2^k$  denotes the binary representation of the integer  $i \in [2^k - 1]$ . In an  $[n = 2^k - 1, k, 2^{k-1}]_2$  punctured Hadamard code, we have  $c_i + c_{2^j} = c_{i+2^j}$ , where  $1 \leq j \leq k - 1$  and  $i \in [2^j - 1]$ . Moreover, we note that an  $[n = 2^k - 1, k, 2^{k-1}]_2$  punctured Hadamard code has a particular structural property: for any  $2 \leq \tilde{k} < k$ , the prefix of length  $2^{\tilde{k}} - 1$  of each codeword is a codeword of the  $[\tilde{n} = 2^{\tilde{k}} - 1, \tilde{k}, 2^{\tilde{k}-1}]_2$  punctured Hadamard code which encodes the message  $(m_1, m_2, \dots, m_{\tilde{k}})$ . We now present the main result of this subsection:

**Lemma 1.** *In an  $[n = 2^k - 1, k, 2^{k-1}]_2$  punctured Hadamard code, any  $1 \leq \ell \leq \frac{n-1}{2}$  erasures can be corrected by contacting at most  $\ell + 1$  other code symbols.*

*Proof:* We prove the Lemma by using induction over  $k$ . For base case, we consider  $k = 2$ , where the  $[n = 3 = 2^2 - 1, 2, 2]_2$  punctured Hadamard code encodes the message  $(m_1, m_2)$  to a codeword  $(c_1, c_2, c_3) = (m_1, m_2, m_1 + m_2)$ . In this case any  $1 \leq \ell \leq \frac{3-1}{2} = 1$  erasure can be recovered by contacting other  $\ell + 1 = 2$  code symbols. For example, one can recover  $c_2 = m_2$  from  $(c_1, c_3) = (m_1, m_1 + m_2)$ .

For inductive step, we assume that the Lemma holds for any punctured code of dimension up to  $k - 1$ . Consider the  $[n = 2^k - 1, k, 2^{k-1}]_2$  punctured Hadamard code of dimension  $k$ , and two cases regarding the positions of  $\ell$  erased code symbols.

- **Case 1: There are  $x \leq 2^{k-2} - 1$  erasures among the first  $\hat{n} = 2^{k-1} - 1$  code symbols.** Note that the first  $\hat{n} = 2^{k-1} - 1$  code symbols constitute a codeword of an  $[\hat{n} = 2^{k-1} - 1, k - 1, 2^{k-2}]_2$  punctured Hadamard code. Therefore, from the inductive hypothesis, one can correct the  $x$  erasures among the first  $\hat{n}$  code symbols by contacting  $x + 1$  other code symbols out of these  $\hat{n}$  code symbols. Now, if the symbol  $c_{2^{k-1}}$  in erasure, we can recover it by contacting one of the unerased symbol among  $\{c_{2^{k-1}+1}, c_{2^{k-1}+2}, \dots, c_{n=2^k-1}\}$  say  $c_{2^{k-1}+j}$  and the corresponding code symbol  $c_j$  from the first  $\hat{n}$  code symbols. Now, we can repair the remaining erased symbols among  $\{c_{2^{k-1}+1}, c_{2^{k-1}+2}, \dots, c_{n=2^k-1}\}$  from  $c_{2^{k-1}}$  and the corresponding code symbol among the first  $\hat{n}$  code symbols. For example, if we want to recover the symbol  $c_{2^{k-1}+m}$ , we can use  $c_{2^{k-1}}$  and  $c_m$  to reconstruct

$c_{2^{k-1}+m}$ . In the worst case, we contact  $\ell + 1$  code symbols during the repair of all  $\ell$  erasures.

- **Case 2: There are  $x \geq 2^{k-2}$  erasures among the first  $\hat{n} = 2^{k-1} - 1$  code symbols.** In this case, we first recover the code symbol  $c_{2^{k-1}}$ , if it is in erasure. Without loss of generality we assume that  $c_{2^{k-1}}$  is in erasure. Note that there are  $\frac{n-1}{2} = 2^{k-1}$  distinct pairs of code symbols  $\{c_i, c_{2^{k-1}+i}\}_{i \in [2^{k-1}]}$  that can recover  $c_{2^{k-1}}$ . Since we have at most  $\frac{n-1}{2} - 1 = 2^{k-1} - 1$  erasures apart from  $c_{2^{k-1}}$ , one of the  $2^{k-1}$  pairs  $\{c_i, c_{2^{k-1}+i}\}_{i \in [2^{k-1}]}$  must be intact. This pair allows us to recover  $c_{2^{k-1}}$ .

Now that we know the symbol  $c_{2^{k-1}} = m_k$ , we can remove the contribution of  $m_k$  from any of the last  $2^k - 1 - 2^k$  code symbols  $\{c_{2^{k-1}+1}, c_{2^{k-1}+2}, \dots, c_{n=2^k-1}\}$ . Similarly, we can add  $m_k$  to any of the first  $\hat{n} = 2^{k-1} - 1$  code symbols  $\{c_1, c_2, \dots, c_{2^{k-1}}\}$ . Therefore, we can reduce the Case 2 to Case 1 of the proof, and repair any  $\ell_1$  erasures by contacting at most  $\ell_1 + 1$  code symbols.

Combining both cases completes the proof.  $\blacksquare$

### E. Generalization of Partition Code Approach

In the construction of the Partition codes, as described in Section IV-A, we use an  $(\frac{\tau}{\ell} + \ell, \frac{\tau}{\ell})$  MDS code to encode disjoint groups of  $\frac{\tau}{\ell}$  message symbols. Note that the rate of this MDS code governs the rate of the overall code.

One can potentially use another code  $\mathcal{C}^{\text{local}}$  of minimum distance at least  $\ell + 1$  to encode disjoint groups of  $\frac{\tau}{\ell}$  message symbols. Now, we use  $r(x)$ ,  $x \in [\ell]$  to denote the number of symbols that needs to be contacted to repair  $x$  erasure in one local group. For the case when an  $(\frac{\tau}{\ell} + \ell, \frac{\tau}{\ell})$  MDS code is used, we have  $r(x) = \frac{\tau}{\ell}$  for  $x \in [\ell]$ . Let  $r^*(x)$  denote the upper concave envelope of  $r(x)$  on the interval  $[1, \ell] \in \mathbb{R}$ . Assume that we have  $g$  disjoint local groups, then a pattern of  $\ell$  erasures can be represented by a vector  $(l_1, l_2, \dots, l_g)$ . Here,  $l_i$  denotes the number of erasures within  $i$ -th local group. Note that we have  $\sum_{i=1}^g l_i = \ell$ .

For a given local code  $\mathcal{C}^{\text{local}}$ , one needs to access  $\sum_{i=1}^g r(l_i)$  number of intact code symbols to repair the erasure pattern  $(l_1, l_2, \dots, l_g)$ . Now, we use concavity of  $r^*(\cdot)$ , the fact that  $r^*(x) \geq r(x)$  for  $x \in [\ell]$ , and Jensen's inequality to obtain the

following.

$$\begin{aligned} \sum_{i=1}^g r(l_i) &\leq \sum_{i=1}^g r^*(l_i) \leq gr^*\left(\frac{\sum_{i=1}^g l_i}{g}\right) \\ &= gr^*\left(\frac{\ell}{g}\right) \end{aligned} \quad (13)$$

Since the rate of the Partition code is agnostic to the number of local groups, we can use the value of  $g$  which minimizes the R.H.S. of (13). This approach optimizes the value of  $r$  for a given choice of  $\ell$  and  $\mathcal{C}^{\text{local}}$ .

#### F. Comment of Random Erasures

The constructions in Section IV-A & IV-B are designed to allow for the cooperative local repairs in the case of adversarial erasure patterns. One can consider the setting where erasures occur according to a random model. Here, we briefly comment on the setting where  $\ell$  erasures are uniformly distributed among the code symbols. Moreover, we assume  $r$  and  $\ell$  to be large enough. We consider the construction of Section IV-A, that is, the Partition codes. In this case, with reasonably high probability (depending on  $r$  and  $\ell$ ), each local group (a total  $g$  of them) experiences about  $t \equiv \Theta\left(\frac{\ell}{g}\right)$  number of erasures. Therefore, with high probability, one can perform cooperative local repair of  $\ell$  random erasures even if an  $\left(\frac{r}{\ell} + t, \frac{r}{\ell}\right)$ -MDS code is employed in the construction of the Partition code (cf. Section IV-A). This translates to a coding scheme with the overall rate of  $\frac{r}{r+t\ell}$ . One can take  $g$  large enough to optimize this value. Indeed, using the techniques of [16], it can be shown that, for random erasures, the partition codes are asymptotically optimal.

#### REFERENCES

- [1] A. Dimakis, P. Godfrey, Y. Wu, M. Wainwright, and K. Ramchandran. Network coding for distributed storage systems. *IEEE Trans. Inf. Theory*, 56(9):4539–4551, 2010.
- [2] K. Rashmi, N. Shah, and P. Kumar. Optimal exact-regenerating codes for distributed storage at the MSR and MBR points via a product-matrix construction. *IEEE Trans. Inf. Theory*, 57:5227–5239, 2011.
- [3] I. Tamo, Z. Wang, and J. Bruck. Zigzag codes: Mds array codes with optimal rebuilding. *IEEE Trans. Inf. Theory*, 59(3):1597–1616, 2013.
- [4] D. Papailiopoulos, A. Dimakis, and V. Cadambe. Repair optimal erasure codes through hadamard designs. *IEEE Trans. Inf. Theory*, 59(5):3021–3037, 2013.
- [5] P. Gopalan, C. Huang, H. Simitci, and S. Yekhanin. On the locality of codeword symbols. *IEEE Trans. Inf. Theory*, 58(11):6925–6934, 2012.
- [6] D. Papailiopoulos and A. Dimakis. Locally repairable codes. In *Proc. of IEEE ISIT*, 2012.
- [7] A. S. Rawat, O. O. Koyluoglu, N. Silberstein, and S. Vishwanath. Optimal locally repairable and secure codes for distributed storage systems. *IEEE Trans. Inf. Theory*, 60(1):212–236, 2014.
- [8] G. Kamath, N. Prakash, V. Lalitha, and P. Kumar. Codes with local regeneration. *CoRR*, abs/1211.1932, 2012.
- [9] I. Tamo and A. Barg. A family of optimal locally recoverable codes. *CoRR*, abs/1311.3284, 2013.
- [10] N. Le Scouarnec A.-M. Kermarrec and G. Straub. Repairing multiple failures with coordinated and adaptive regenerating codes. In *Proc. International Symposium on Network Coding*, 2011.
- [11] K. W. Shum and Y. Hu. Cooperative regenerating codes. *IEEE Trans. Inf. Theory*, 59(11):7229–7258, 2013.
- [12] F. Oggier and A. Datta. Self-repairing homomorphic codes for distributed storage systems. In *Proc. of INFOCOM*, 2011.
- [13] N. Prakash, G. Kamath, V. Lalitha, and P. Kumar. Optimal linear codes with a local-error-correction property. In *Proc. of IEEE ISIT*, 2012.

- [14] V. R. Cadambe and A. Mazumdar. An upper bound on the size of locally recoverable codes. *CoRR*, abs/1308.3200, abs/1308.3200, 2013.
- [15] M. Forbes and S. Yekhanin. On the locality of codeword symbols in non-linear codes. *CoRR*, abs/1303.3921, 2013.
- [16] A. Mazumdar, V. Chandar, and G. W. Wornell. Local recovery properties of capacity achieving codes. In *Information Theory and Applications Workshop (ITA)*, 2013, pages 1–3. IEEE, 2013.

#### APPENDIX A

##### PART OF THE PROOF OF THEOREM 1

For the construction of a subcode as described in Fig. 1, we define  $\mathcal{A}_j = \mathcal{I}_j \setminus \mathcal{I}_{j-1} \subseteq \mathcal{R}_j \cup \{i_1^j, \dots, i_\ell^j\}$  and  $a_j = |\mathcal{A}_j|$ . Assuming that the while loop in Fig. 1 ends with  $j = t$ , for  $j \in [t]$ , we have

$$\mathcal{I}_j = \bigcup_{j' \in [j]} \mathcal{A}_{j'},$$

where we take union of the disjoint sets

$$\mathcal{A}_{j'}, \quad j' \in [j].$$

Note that the code symbols indexed by the set  $\{i_1^j, \dots, i_\ell^j\}$  are functions of the code symbols indexed by the set  $\mathcal{R}_j = \Gamma_{\{i_1^j, \dots, i_\ell^j\}}$ . Therefore, at line 7, there are at most  $q^{a_j - \ell}$  possibilities for  $\mathbf{y}_j$ . This implies that

$$|\mathcal{C}_j| \geq |\mathcal{C}_{j-1}|/q^{a_j - \ell}. \quad (14)$$

The construction of the subcode  $\mathcal{C}'$  can end at either line 11 or line 15. Here we analyze only the case when the construction ends at line 11. In this case, we have  $|\mathcal{C}_t| \leq q$ , or

$$1 \geq \log_q |\mathcal{C}_t| \geq k - \sum_{j=0}^{t-1} (a_{j+1} - \ell). \quad (15)$$

Now, using that  $a_j \leq |\mathcal{A}_j| \leq |\mathcal{R}_j \cup \{i_1^j, \dots, i_\ell^j\}| \leq r + \ell$ , we get

$$k - 1 \leq \sum_{j=0}^{t-1} (a_{j+1} - \ell) \leq tr, \quad (16)$$

which gives us that

$$t \geq \left\lceil \frac{k}{r} \right\rceil - 1. \quad (17)$$

Note that sub-code  $\mathcal{C}' = \mathcal{C}_t$ . Therefore,

$$\begin{aligned} \log_q |\mathcal{C}'| &= \log_q |\mathcal{C}_t| \\ &\geq \log_q |\mathcal{C}| - \sum_{j=0}^{t-1} (a_{j+1} - \ell) \\ &= k - \sum_{j=0}^{t-1} a_{j+1} + t\ell \\ &\stackrel{(a)}{=} k - |\mathcal{I}_t| + t\ell \end{aligned} \quad (18)$$

where (a) follows from the fact that  $\mathcal{I}_t$  is union of the disjoint sets  $\mathcal{A}_j$ .