ϵ-MSR Codes with Small Sub-packetization

Ankit Singh Rawat[†], Itzhak Tamo[‡], Venkatesan Guruswami[§], and Klim Efremenko[‡]

[†] Massachusetts Institute of Technology, Cambridge, MA 02139, USA,

[‡] Tel Aviv University, Tel Aviv, Israel,

[§] Carnegie Mellon University, Pittsburgh, PA 15213, USA.

E-mail: asrawat@mit.edu, tamo@post.tau.ac.il, venkatg@cs.cmu.edu, klimefrem@gmail.com.

Abstract

Minimum storage regenerating (MSR) codes form a special class of maximum distance separable (MDS) codes by providing mechanisms for exact regeneration of a single code block in their codewords by downloading the minimum amount of information from the remaining code blocks. As a result, the MSR codes find application to distributed storage systems to enable node repairs with the optimal repair bandwidth. However, the construction of exact-repairable MSR codes requires working with a large sub-packetization level, which restricts the employment of these codes in practice. This paper explores exact-repairable MDS codes that significantly reduce the required sub-packetization level by achieving slightly sub-optimal repair bandwidth as compared to the MSR codes. This paper presents a general approach to combine an MSR code with large sub-packetization level with a code with large enough minimum distance to construct exact-repairable MDS codes with small sub-packetization level and near-optimal repair bandwidth. For a given number of parity blocks, the codes constructed using this approach have their sub-packetization level scaling logarithmically with the code length. In addition, the obtained codes require field size linear in the code length and ensure load balancing among the intact code blocks in terms of the information downloaded from these blocks during a node repair.

I. Introduction

Maximum distance separable (MDS) codes are among the most preferred codes to be employed in distributed storage systems. However, the sustained applicability of these codes in distributed storage systems depends on the availability of low-cost mechanisms to exactly regenerate parts of their codewords to be able to replenish the redundancy lost due to node failures [1]. Such mechanisms also enable efficient access to the content of a temporarily unavailable storage node with the help of the data stored on the remaining (available) nodes.

In [2], Dimakis et al. introduce repair bandwidth – the amount of data downloaded to repair a failed node – as a metric to measure the efficiency of a repair scheme. In particular, they consider a setup where a coding scheme encodes a file containing k symbols over a finite field $\mathbb{F}$ to a codeword comprising n symbols over $\mathbb{F}$. These n symbols are then stored on n storage nodes. For the repair of a failed node, each code symbol of the codeword is viewed as an ℓ length vector (code block) over a subfield $\mathbb{B}$, where ℓ is referred to as *sub-packetization level* or *node size*. Assuming that the underlying code is an MDS code and t out of $n-1$ intact nodes are contacted to regenerate the code block stored on the failed node, it is necessary to download at least [2], [3]

$$\frac{t}{t-k+1} \cdot \ell \quad \text{symbols (over } \mathbb{B}\text{).} \tag{1}$$

The problem of designing exact-repairable MDS codes with the optimal repair bandwidth (cf. (1)) has led to many novel code designs that are proposed in [4]–[6] and references therein. In [7], Ye and Barg present the first fully explicit construction of MDS codes with optimal repair bandwidth for all values of the system parameters n, k, and t. Further explicit constructions of MDS codes with optimal repair bandwidth are presented in [8], [9]. These optimal constructions also ensure load balancing as the same amount of data is downloaded from each of the t contacted nodes during the repair of a failed node. In the literature, such codes are referred to as *minimum storage regenerating (MSR)* codes [2].

For the low rate setting, i.e., $2k-2 \leq t \leq n-1$, the MSR codes proposed in [4] have the minimum possible sub-packetization level $\ell = t-k+1$. However, the existing constructions of high rate MSR codes need to work with large sub-packetization levels. In particular, the constructions presented in [8], [9] require $\ell = (n-k)^{\left\lceil \frac{n}{n-k} \right\rceil}$ for $t = n-1$. Note that this sub-packetization level increases with the rate of the code. Specifically, the sub-packetization level ℓ becomes exponential in the code length n for $n-k = \Theta(1)$. The MSR codes with similar scaling for sub-packetization level for general values of $k \leq t \leq n-1$, i.e., $\ell = (t-k+1)^{\left\lceil \frac{n}{t-k+1} \right\rceil}$, are presented in [10]. On the converse side, for the setting with $t = n-1$, Goparaju et al. establish the following lower bound on the sub-packetization level of an MSR code [11].

$$k \leq 2 \cdot (\log_2 \ell) \left(\left\lfloor \log_{\frac{n-k}{n-k-1}} \ell \right\rfloor + 1 \right). \tag{2}$$

Note that, for $n-k = \Theta(1)$, the bound in (2) implies that $\ell = \Omega\big(\exp(\sqrt{k})\big)$. Thus, for the setting with constant number of parity symbols, an MSR code necessarily has a very large sub-packetization level. This restricts the use of the MSR codes in practical systems as large sub-packetization levels reduce the design space for storage providers [11]. In addition, the MSR codes with large sub-packetization levels do not provide bandwidth efficient access to the parts of the information stored on the system by using degraded reads [12].

In this paper, we relax the requirement that the underlying MDS code attains the optimal repair bandwidth. In particular, we design MDS codes with repair bandwidth arbitrarily close to the bound given in (1). This small loss in terms of repair bandwidth optimality results in a significant benefit in terms of the sub-packetization level. In particular, we present constructions of MDS codes with constant number of parity symbols and near optimal repair bandwidth while working with the sub-packetization level that scales only *logarithmically* with the code length. This amounts to a doubly exponential saving in terms of the sub-packetization level as compare to the existing MSR codes [7], [9]. Similarly to the MSR codes, we also require that the repair mechanism of our designed codes ensure load balancing among the contacted nodes. We name the proposed codes as ϵ-MSR codes: For a given $\epsilon \geq 0$, an ϵ-MSR code downloads at most $(1+\epsilon) \cdot \ell/(t-k+1)$ symbols (over $\mathbb{B}$) from each of the t contacted nodes during the repair of a failed node. Throughout this paper, we restrict ourselves to $t = n-1$ case. Similar results can be easily obtained for the settings with other values of t as well.

Related work: The problem of designing exact-repairable MDS codes with both small repair bandwidth and small sub-packetization level has been previously explored in [5], [13]. After the presentation of the initial results of this paper [14], Guruswami and Rawat have also studied this problem in [15]. We note that the work in [5], [13] enable efficient repair for only a set of k nodes. We also note that even though the work in [15] realizes near optimal

repair bandwidth for all nodes, similar to [5], [13], this work does not ensure load balancing among the contacted nodes. Furthermore, unlike the work in [5], [15], the codes presented in this paper require field size scaling linearly in the code length. We present a detailed comparison with [15] in Remark 4 (cf. Sec. IV).

Organization: In Sec. II, we present the necessary background on linear array codes and their linear repair schemes. The main result of this paper is a general approach which transforms a short MSR code with large sub-packetization level to a long linear array code with both small repair bandwidth and small sub-packetization level. We describe this general approach in Sec. III. In Sec. IV, we utilize a code construction from [7] to generate the short MSR code in our general approach so that the obtained long code is an MDS code; thus, giving us an ϵ-MSR code. In Sec. V, we explore the sub-packetization level that is necessary to realize an ϵ-MSR code. We conclude the paper in Sec. VI with some directions for future work.

II. Preliminaries

For an integer $a > 0$, we use $[a]$ to denote the set $\{1, 2, \ldots, a\}$. Similarly, for two integers a and b such that $a \leq b$, $[a : b]$ represents the set $\{a, a+1, \ldots, b\}$.

A. Linear array codes

Let $\mathbb{F}$ be a finite field which is the degree ℓ extension of its subfield $\mathbb{B}$. We say that a set $\mathcal{C} \subseteq \mathbb{F}^n$ forms an $(n, M, d_{\min})_{\mathbb{F}}$ code, if we have $|\mathcal{C}| = M$ and

$$d_{\min} = \min_{\mathbf{c} \neq \mathbf{c}' \in \mathcal{C}} d_H(\mathbf{c}, \mathbf{c}'),$$

where $d_H(\cdot, \cdot)$ denotes the Hamming distance. Note that each element of $\mathbb{F}$ can be represented as an ℓ-length vector over $\mathbb{B}$. Therefore, we can express a codeword $\mathbf{c} = (c_1, \ldots, c_n) \in \mathcal{C} \subseteq \mathbb{F}^n$ as an $n\ell$-length vector $\mathbf{c} = (\mathbf{c}_1, \ldots, \mathbf{c}_n) \in \mathbb{B}^{n\ell}$. Here, for $i \in [n]$, the *code block* $\mathbf{c}_i = (c_{i,1}, \ldots, c_{i,\ell}) \in \mathbb{B}^{\ell}$ denotes the ℓ-length vector corresponding to the *code symbol* $c_i \in \mathbb{F}$.

In this equivalent representation, we say that $\mathcal{C}$ is a *linear array code* if it forms a $\log_{|\mathbb{B}|} M$-dimensional subspace (over $\mathbb{B}$) of $\mathbb{B}^{n\ell}$. Moreover, we refer to the code as an $[n, \log_{|\mathbb{B}|} M, d_{\min}, \ell]_{\mathbb{B}}$ linear array code. We say that an $[n, \log_{|\mathbb{B}|} M, d_{\min}, \ell]_{\mathbb{B}}$ linear array code is a *maximum distance separable* (MDS) code if ℓ divides $\log_{|\mathbb{B}|} M$ and

$$d_{\min} = n - \left(\log_{|\mathbb{B}|} M\right)/\ell + 1.$$

An $[n, \log_{|\mathbb{B}|} M, d_{\min}, \ell]_{\mathbb{B}}$ linear array code can be defined by an $(n\ell - \log_{|\mathbb{B}|} M) \times n\ell$ full rank matrix $\mathbf{H}$ over $\mathbb{B}$ as follows.

$$\mathcal{C} = \left\{\mathbf{c} = (\mathbf{c}_1, \ldots, \mathbf{c}_n) \ : \ \mathbf{H} \cdot \mathbf{c} = 0\right\} \subseteq \mathbb{B}^{n\ell}. \tag{3}$$

The matrix $\mathbf{H}$ is called the *parity check matrix* of the code $\mathcal{C}$. Assuming that k is an integer such that $\log_{|\mathbb{B}|} M = k\ell$, the parity check matrix $\mathbf{H}$ can be viewed as a block matrix

$$\mathbf{H} = \begin{pmatrix} \mathbf{H}_1 & \mathbf{H}_2 & \cdots & \mathbf{H}_n \end{pmatrix} \in \mathbb{B}^{(n-k)\ell \times n\ell}. \tag{4}$$

For $i \in [n]$, we refer to the $(n-k)\ell \times \ell$ sub-matrix $\mathbf{H}_i$ as the *thick column* associated with the i-th code block in the codewords of $\mathcal{C}$.

B. Linear repair schemes for a linear array code

Let $\mathbf{c} = (\mathbf{c}_1, \ldots, \mathbf{c}_n)$ be a codeword in the linear array code $\mathcal{C}$ defined by the parity check matrix $\mathbf{H}$. Recall that we are considering a distributed storage setup where these n code blocks are stored on n distinct storage nodes. For every $i \in [n]$, we are interested in the task of repairing (regenerating) the code block $\mathbf{c}_i$ by downloading a small number of symbols (over $\mathbb{B}$) from the $n-1$ nodes storing the remaining code blocks $\{\mathbf{c}_j\}_{j \neq i}$. A *linear repair scheme* performs this task with the help of linear operations over the base field $\mathbb{B}$. We summarize the description of a linear repair scheme and the associated repair bandwidth in the following statement.

Proposition II.1. *Let $S_i \in \mathbb{B}^{\ell \times (n-k)\ell}$ be a matrix such that the following two conditions hold.*

$$\text{rank}\,(S_i \mathbf{H}_i) = \ell \tag{5}$$

and

$$\sum_{j \in [n] \setminus \{i\}} \text{rank}\,(S_i \mathbf{H}_j) \leq \gamma. \tag{6}$$

Then, the code block $\mathbf{c}_i$ can be regnenerated by downloading at most γ symbols (over $\mathbb{B}$) from the remaining $n-1$ node. In particular, this requires downloading $\text{rank}\,(S_i \mathbf{H}_j)$ *symbols (over $\mathbb{B}$) from the node storing the code block $\mathbf{c}_j$.*

Proof. Given the matrix S_i, one can regenerate the code block $\mathbf{c}_i$ by downloading at most γ symbols (over $\mathbb{B}$) from the remaining $n-1$ nodes as follows.

- Note that the codeword $\mathbf{c} = (\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_n)$ satisfies the following.

$$\mathbf{H}\mathbf{c} = \mathbf{H}_1 \mathbf{c}_1 + \cdots + \mathbf{H}_n \mathbf{c}_n = 0 \tag{7}$$

- By multiplying (7) from left by the matrix S_i which satisfies (5) and (6), we obtain

$$S_i \mathbf{H}_i \mathbf{c}_i = - \sum_{j \in [n] \setminus \{i\}} S_i \mathbf{H}_j \mathbf{c}_j \tag{8}$$

Note that in order to evaluate the right hand side of (8), for $j \in [n]\setminus\{i\}$, we need to download at most $\text{rank}\,(S_i \mathbf{H}_j)$ symbols (over $\mathbb{B}$) from the node storing the code block $\mathbf{c}_j$. It follows from (6) that we download at most γ symbols from the code blocks $\{\mathbf{c}_1, \ldots, \mathbf{c}_{i-1}, \mathbf{c}_{i+1}, \ldots, \mathbf{c}_n\}$. Once we know the right hand side of (8), we can solve for $\mathbf{c}_i$ as it follows from (6) that the matrix $S_i \mathbf{H}_i$ is full rank. □

C. ϵ-MSR codes

The MSR codes [2] ensure that for every $i \in [n]$, it is possible to repair the i-th code block by downloading exactly $\frac{\ell}{n-k}$ symbols (over $\mathbb{B}$) from each of the remaining intact nodes. We now formally define the ϵ-MSR codes. For brevity, we consider only linear array codes with linear repair schemes.

Definition 1. (ϵ-MSR code): *Let $\mathcal{C}$ be an $[n, k\ell, d_{\min} = n-k+1, \ell]_{\mathbb{B}}$ MDS code. For $\epsilon > 0$, we say that the code $\mathcal{C}$ is an $(n, k, t = n-1, \ell)_{\mathbb{B}}$ ϵ-MSR code if, for every $i \in [n]$, there is a linear repair scheme to repair the i-th code block $\mathbf{c}_i$ with*

$$\beta_{j,i} \leq (1+\epsilon) \cdot \frac{\ell}{n-k} \text{ symbols (over } \mathbb{B}) \quad \forall\, j \in [n]\setminus\{i\}.$$

Here, $\beta_{j,i}$ denotes the number of symbols that the code block $\mathbf{c}_j$ contributes during the repair of the code block $\mathbf{c}_i$.

Remark 1. Since an $(n, k, t = n-1, \ell)_{\mathbb{B}}$ ϵ-MSR code with $\epsilon = 0$ is an MSR code, we simply refer to it as an $(n, k, t = n-1, \ell)_{\mathbb{B}}$ MSR code.

III. Repair-efficient Linear Array Codes with Small Sub-packetization Levels

In this section, we present a general approach to realize our end goal of constructing ϵ-MSR codes with small sub-packetization levels. Towards this, we combine a short MSR code with another code that have large enough distance to obtain a linear array code which has a small sub-packetization level as compared to its length. In particular, for a constant number of parity blocks, it is possible to obtain codes that have the code length exponential in their sub-packetization levels. Furthermore, the repair bandwidth of the obtained code is only slightly larger than the lower bound on the repair bandwidth of an MDS code with the same parameters (cf. (1)). In Sec. IV, we utilize a family of MSR codes from [7] to ensure that the obtained long code is an MDS code. As a result, the approach described in this section gives us ϵ-MSR codes with small sub-packetization levels.

Construction 1. We are given two codes.

1) An $(n = k + r, k, t = n-1, \ell)_{\mathbb{B}}$ MSR code $\mathcal{C}^{\mathrm{I}}$ defined by the parity check matrix

$$\mathbf{H} = \begin{pmatrix} H_{1,1} & H_{1,2} & \cdots & H_{1,n} \\ \vdots & \vdots & \ddots & \vdots \\ H_{r,1} & H_{r,2} & \cdots & H_{r,n} \end{pmatrix} \in \mathbb{B}^{r\ell \times n\ell}. \tag{9}$$

For $i \in [n]$, the repair matrix associated with the i-th code block takes the following diagonal form.

$$S_i = \mathrm{Diag}\,(S_{i,1}, S_{i,2}, \ldots, S_{i,r}) \in \mathbb{B}^{\ell \times r\ell}, \tag{10}$$

where for each $j \in [r]$, $S_{i,j}$ is an $\frac{\ell}{r} \times \ell$ matrix (over $\mathbb{B}$).

2) An $(N, M, D)_{\mathbb{G}}$ code $\mathcal{C}^{\mathrm{II}}$ over the alphabet $\mathbb{G}$ of size at most n.

Given these two codes, we construct an $[\mathcal{N} = M, \mathcal{K}l = (M-r)l, \mathcal{D}, l = N\ell]_{\mathbb{B}}$ linear array code $\mathcal{C} = \mathcal{C}^{\mathrm{II}} \circ \mathcal{C}^{\mathrm{I}}$ by designing its $rN\ell \times MN\ell$ parity check matrix $\mathcal{H}$. Note that a codeword of $\mathcal{C}$ comprises $M = |\mathcal{C}^{\mathrm{II}}|$ code blocks with each of these blocks containing $N\ell$ symbols (over $\mathbb{B}$). The M code blocks in a codeword of $\mathcal{C}$ are indexed by M distinct N-length codewords in $\mathcal{C}^{\mathrm{II}}$. Let $\mathbf{c} = (c_1, \ldots, c_N) \in \mathbb{G}^N$ be a codeword of $\mathcal{C}^{\mathrm{II}}$. Then, the $N\ell$ colums of the parity check matrix $\mathcal{H}$ that correspond to the code block of a codeword of $\mathcal{C}$ indexed by $\mathbf{c} \in \mathcal{C}^{\mathrm{II}}$ are defined as follows.

$$\mathcal{H}_{\mathbf{c}} = \begin{bmatrix} \alpha_{1,\mathbf{c}} \cdot \mathrm{Diag}(H_{1,c_1}, \ldots, H_{1,c_N}) \\ \vdots \\ \alpha_{r,\mathbf{c}} \cdot \mathrm{Diag}(H_{r,c_1}, \ldots, H_{r,c_N}) \end{bmatrix}, \tag{11}$$

where $\{\alpha_{j,\mathbf{c}}\}_{j \in [r], \mathbf{c} \in \mathcal{C}^{\mathrm{II}}}$ are non-zero elements from $\mathbb{B}$. We associate the alphabet $\mathbb{G}$ with the set $[|\mathbb{G}|]$ while specifying the parity check matrix $\mathcal{H}$ in (11). Note that all the blocks $\{H_{j,c_i}\}_{j \in [r], i \in [N]}$ in (11) are well defined as we have $|\mathbb{G}| \leq n$.

As shown in Sec. IV, depending on the specific choice of the MSR code $\mathcal{C}^{\mathrm{I}}$, these scalars can be chosen to ensure that the obtained code $\mathcal{C}$ is an MDS code. Next, we show that the code $\mathcal{C}$ obtained from Construction 1 has a linear repair scheme with small repair bandwidth, regardless of the choice for these scalars.

Theorem III.1. *Let $\mathcal{C}^{\mathrm{I}}$ be an $(n = k + r, k, t = n - 1, \ell)_{\mathbb{B}}$ MSR code with the $r\ell \times n\ell$ parity check matrix $\mathbf{H}$ (cf. (9)) and $\mathcal{C}^{\mathrm{II}}$ be an $(N, M, D = \delta N)_{\mathbb{G}}$ code with $|\mathbb{G}| \leq n$. Let the block diagonal matrices $\left\{S_i\right\}_{i\in[n]}$ (cf. (10)) enable repair bandwidth optimal repairs for $\mathcal{C}^{\mathrm{I}}$. Then, the code $\mathcal{C} = \mathcal{C}^{\mathrm{II}} \circ \mathcal{C}^{\mathrm{I}}$ as defined in Construction 1 is an $[\mathcal{N} = M, (\mathcal{N} - r)N\ell, \mathcal{D}, N\ell]_{\mathbb{B}}$ linear array code which enables repair of every code block in each of its codewords by downloading at most*

$$\big(1 + (r-1)(1-\delta)\big) \cdot N\ell/r$$

symbols (over $\mathbb{B}$) from each of the remaining $\mathcal{T} = \mathcal{N} - 1$ code blocks of the associated codeword.

Proof. For $i \in [M]$, we demonstrate a linear repair scheme for the i-th code block of a codeword of $\mathcal{C}$. Recall that the M code blocks in a codeword of $\mathcal{C}$ are indexed by M distinct codewords in the code $\mathcal{C}^{\mathrm{II}}$. Let the code block to be repaired be indexed by the codeword $\mathbf{c} = (c_1, c_2, \ldots, c_N) \in \mathcal{C}^{\mathrm{II}}$. We claim that the following $N\ell \times rN\ell$ matrix (over $\mathbb{B}$) serves as a repair matrix for this code block.

$$\mathcal{S}_{\mathbf{c}} = \mathrm{Diag}\left(\mathrm{Diag}(S_{c_1,1}, \ldots, S_{c_N,1}), \cdots, \mathrm{Diag}(S_{c_1,r}, \ldots, S_{c_N,r})\right)$$

It is sufficient to verify the conditions given in (6), i.e.,

$$\mathrm{rank}\left(\mathcal{S}_{\mathbf{c}} \begin{bmatrix} \mathcal{H}_{1,\mathbf{c}} \\ \vdots \\ \mathcal{H}_{r,\mathbf{c}} \end{bmatrix}\right) = N\ell. \tag{12}$$

Recall that, as per our assumption, $\mathbf{c}$ denotes the i-th codeword of the code $\mathcal{C}^{\mathrm{II}}$. Note that we have

$$\begin{aligned}\mathcal{S}_{\mathbf{c}} \begin{bmatrix} \mathcal{H}_{1,\mathbf{c}} \\ \vdots \\ \mathcal{H}_{r,\mathbf{c}} \end{bmatrix} &= \mathrm{Diag}\left(\mathrm{Diag}(S_{c_1,1}, \ldots, S_{c_N,1}), \cdots, \mathrm{Diag}(S_{c_1,r}, \ldots, S_{c_N,r})\right) \begin{pmatrix} \alpha_{1,\mathbf{c}} \cdot \mathrm{Diag}(H_{1,c_1}, \ldots, H_{1,c_N}) \\ \vdots \\ \alpha_{r,\mathbf{c}} \cdot \mathrm{Diag}(H_{r,c_1}, \ldots, H_{r,c_N}) \end{pmatrix} \\ &= \begin{pmatrix} \alpha_{1,\mathbf{c}} \cdot \mathrm{Diag}(S_{c_1,1}H_{1,c_1}, \ldots, S_{c_N,1}H_{1,c_N}) \\ \vdots \\ \alpha_{r,\mathbf{c}} \cdot \mathrm{Diag}(S_{c_1,r}H_{r,c_1}, \ldots, S_{c_N,r}H_{r,c_N}) \end{pmatrix} \end{aligned} \tag{13}$$

Therefore, we have

$$\begin{aligned}\mathrm{rank}\left(\mathcal{S}_{\mathbf{c}} \begin{bmatrix} \mathcal{H}_{1,\mathbf{c}} \\ \vdots \\ \mathcal{H}_{r,\mathbf{c}} \end{bmatrix}\right) &= \sum_{j=1}^{N} \mathrm{rank}\left(\begin{bmatrix} S_{c_j,1}H_{1,c_j} \\ \vdots \\ S_{c_j,r}H_{r,c_j} \end{bmatrix}\right) \\ &\overset{(i)}{=} \sum_{j=1}^{N} \mathrm{rank}\left(S_{c_j}\mathbf{H}_{c_j}\right) \overset{(ii)}{=} \sum_{j=1}^{N} \ell = N\ell, \end{aligned} \tag{14}$$

where the step (i) follows from the structure of the repair matrix S_i in the short MSR code $\mathcal{C}^{\mathrm{I}}$ (cf. (10)). The step (ii) follows from the requirement on the repair matrices of $\mathcal{C}^{\mathrm{I}}$ (cf. (5)).

Repair bandwidth: Next, we focus on the repair bandwidth associated with the repair matrix $\mathcal{S}_{\mathbf{c}}$. For a codeword $\tilde{\mathbf{c}} = (\tilde{c}_1, \tilde{c}_2, \ldots, \tilde{c}_N) \in \mathcal{C}^{\mathrm{II}}$ such that $\tilde{\mathbf{c}} \neq \mathbf{c}$, the code block in a codeword of $\mathcal{C}$ which is indexed by $\widetilde{\mathbf{c}}$ needs to contribute

$$\operatorname{rank}\left(\mathcal{S}_{\mathbf{c}}\begin{bmatrix}\mathcal{H}_{1,\tilde{\mathbf{c}}}\\ \vdots \\ \mathcal{H}_{r,\tilde{\mathbf{c}}}\end{bmatrix}\right)$$

symbols (over $\mathbb{B}$) during the repair of the code block of interest, i.e., the code block indexed by the codeword $\mathbf{c} \in \mathcal{C}^{\mathrm{II}}$. Note that

$$\begin{aligned}\mathcal{S}_{\mathbf{c}}\begin{bmatrix}\mathcal{H}_{1,\tilde{\mathbf{c}}}\\ \vdots \\ \mathcal{H}_{r,\tilde{\mathbf{c}}}\end{bmatrix} &= \operatorname{Diag}\left(\operatorname{Diag}(S_{c_1,1},\ldots,S_{c_N,1}),\cdots,\operatorname{Diag}(S_{c_1,r},\ldots,S_{c_N,r})\right)\begin{pmatrix}\alpha_{1,\tilde{\mathbf{c}}}\cdot\operatorname{Diag}(H_{1,\tilde{c}_1},\ldots,H_{1,\tilde{c}_N})\\ \vdots \\ \alpha_{r,\tilde{\mathbf{c}}}\cdot\operatorname{Diag}(H_{r,\tilde{c}_1},\ldots,H_{r,\tilde{c}_N})\end{pmatrix}\\ &= \begin{pmatrix}\alpha_{1,\tilde{\mathbf{c}}}\cdot\operatorname{Diag}(S_{c_1,1}H_{1,\tilde{c}_1},\ldots,S_{c_N,1}H_{1,\tilde{c}_N})\\ \vdots \\ \alpha_{r,\tilde{\mathbf{c}}}\cdot\operatorname{Diag}(S_{c_1,r}H_{r,\tilde{c}_1},\ldots,S_{c_N,r}H_{r,\tilde{c}_N})\end{pmatrix} \qquad (15)\end{aligned}$$

Therefore, we have

$$\operatorname{rank}\left(\mathcal{S}_{\mathbf{c}}\begin{bmatrix}\mathcal{H}_{1,\tilde{\mathbf{c}}}\\ \vdots \\ \mathcal{H}_{r,\tilde{\mathbf{c}}}\end{bmatrix}\right) = \sum_{j=1}^{N}\operatorname{rank}\left(\begin{bmatrix}S_{c_j,1}H_{1,\tilde{c}_j}\\ \vdots \\ S_{c_j,r}H_{r,\tilde{c}_j}\end{bmatrix}\right) \overset{(\mathrm{i})}{=} \sum_{j=1}^{N}\operatorname{rank}(S_{c_j}\mathbf{H}_{\tilde{c}_j}), \qquad (16)$$

where the step (i) follows from (10). We now consider two cases.

1) **Case 1 ($\widetilde{c}_j = c_j$):** In this case, we have

$$\operatorname{rank}(S_{c_j}\mathbf{H}_{\tilde{c}_j}) = \operatorname{rank}(S_{c_j}\mathbf{H}_{c_j}) \overset{(\mathrm{i})}{=} \ell, \qquad (17)$$

where the steps (i) follows from (5).

2) **Case 2 ($\widetilde{c}_j \neq c_j$):** Note that we have

$$\operatorname{rank}(S_{c_j}\mathbf{H}_{\tilde{c}_j}) \overset{(\mathrm{i})}{=} \frac{\ell}{r}, \qquad (18)$$

where the steps (i) follows from the fact that S_{c_i} is the repair matrix for the c_i-th code block of the MSR code $\mathcal{C}^{\mathrm{I}}$. Recall that we have $c_i \in \mathbb{G}$ is associated with an element of $[|\mathbb{G}|] \subseteq [n]$.

By substituting (17) and (18) in (16), we obtain that

$$\begin{aligned}\operatorname{rank}\left(\mathcal{S}_{\mathbf{c}}\begin{bmatrix}\mathcal{H}_{1,\tilde{\mathbf{c}}}\\ \vdots \\ \mathcal{H}_{r,\tilde{\mathbf{c}}}\end{bmatrix}\right) &= \sum_{j\in[N]:c_j=\tilde{c}_j}\operatorname{rank}(S_{c_j}\mathbf{H}_{\tilde{c}_j}) + \sum_{j\in[N]:c_j\neq\tilde{c}_j}\operatorname{rank}(S_{c_j}\mathbf{H}_{\tilde{c}_j})\\ &= |\{j\in[N]: c_j=\tilde{c}_j\}|\ell + |\{j\in[N]: c_j\neq\tilde{c}_j\}|\frac{\ell}{r}\end{aligned}$$

$$= N\ell - |\{j \in [N] : c_j \neq \tilde{c}_j\}| \left(\frac{r-1}{r}\right)\ell \tag{19}$$

$$\leq N\ell - D\left(\frac{r-1}{r}\right)\ell$$

$$= \frac{N\ell}{r} + \left(\frac{r-1}{r}\right)(N-D)\ell \tag{20}$$

$$\overset{(i)}{=} \big(1 + (r-1)(1-\delta)\big) \cdot \frac{N\ell}{r}, \tag{21}$$

where we use $D = \delta N$ in the step (i). □

Remark 2. For a given $\epsilon > 0$, we can choose the codes $\mathcal{C}^{II}$ to be such that

$$\delta = \frac{D}{N} \geq 1 - \frac{\epsilon}{r-1}.$$

Combining this with (21) gives us that

$$\operatorname{rank}\left(\mathcal{S}_{\mathbf{c}} \begin{bmatrix} \mathcal{H}_{1,\tilde{\mathbf{c}}} \\ \vdots \\ \mathcal{H}_{r,\tilde{\mathbf{c}}} \end{bmatrix}\right) \leq \big(1+\epsilon\big) \cdot \frac{N\ell}{r}. \tag{22}$$

This implies that the repair bandwidth of the code $\mathcal{C} = \mathcal{C}^{II} \circ \mathcal{C}^{I}$ is at most $(1+\epsilon)$ times the lower bound on the repair bandwidth of an MDS code with the same parameters (cf. (1)).

Remark 3. Assuming that the code $\mathcal{C}^{II}$ is a linear code over an alphabet of size q, its average distance satisfies the following.

$$\overline{d} = \frac{1}{|\mathcal{C}^{II}|(|\mathcal{C}^{II}|-1)} \sum_{\mathbf{c},\mathbf{c}' \in \mathcal{C}:\ \mathbf{c} \neq \mathbf{c}'} d_H(\mathbf{c},\mathbf{c}') = \frac{(q-1)|\mathcal{C}^{II}|}{q(|C^{II}|-1)} N \geq (1-1/q)\, N.$$

This can be combined with (19) to conclude that the obtained linear array code downloads on average $(1 + (r-1)/q) \cdot \frac{N\ell}{r}$ symbols (over $\mathbb{B}$) from an intact code block.

In Table I, we illustrate the repair bandwidth of a linear array code obtained by Construction 1 with the help of a few examples. We employ the MSR codes obtained in [16] as the short codes $\mathcal{C}^{I}$ in these examples. Moreover, all the code used as $\mathcal{C}^{II}$ in these examples are linear codes. This allows us to obtain upper bound on the average number of symbols downloaded from a code block stored on an intact node (cf. Remark 3).

IV. Long MDS Codes with Near-optimal Repair Bandwidth

In this section, we utilize a specific family of MSR codes from [7] to instantiate the short code $\mathcal{C}^{I}$ in our general approach described in Construction 1. This choice ensures that the long code obtained from the construction is an MDS codes. For a given $\epsilon > 0$, this allows us to obtain a construction for ϵ-MSR codes by employing a code with large enough distance as $\mathcal{C}^{II}$ (cf. Remark 2). Before, we present the family of ϵ-MSR codes, we briefly describe the construction by Ye and Barg [7] along with the associated repair scheme.

$\mathcal{C}^{\mathrm{I}}$ $(n, k, t = n-1, \ell)$ MSR code	$\mathcal{C}^{\mathrm{II}}$ $(N, K, D = \delta N)_q$ code	$\mathcal{N} = q^K$	r	$\mathcal{K} = q^K - r$	β/l	$\overline{\beta}/l$	$l = N\ell = N \cdot r^{\frac{q}{r+1}}$
$(3, 1, 2, 2)$ MSR code	$[20, 3, 13]_3$	27	2	25	0.675	0.653	40 (2^9)
$(9, 7, 8, 8)$ MSR code	$[10, 2, 9]_9$	81	2	79	0.55	0.55	80 (2^{27})
$(9, 7, 8, 8)$ MSR code	$[15, 3, 12]_9$	729	2	727	0.6	0.554	120 (2^{81})
$(8, 5, 7, 9)$ MSR code	$[20, 3, 16]_8$	512	3	509	0.466	0.415	180 (3^{128})

TABLE I: Examples of linear array codes obtained using Construction 1. These short codes $\mathcal{C}^{\mathrm{I}}$ utilized in these examples are constructed by Wang et al. [16]. β and $\overline{\beta}$ denote the upper bounds on the maximum number of symbol and average number of symbols downloaded from an intact code block during the repair process, respectively. The term inside the brackets in the rightmost column represents the sub-packetization level needed by the best known constructions for the MSR codes with parameters $\mathcal{N}, \mathcal{K}$ and $\mathcal{D} = \mathcal{N} - 1$.

A. Ye and Barg construction [7]

Let $\mathbb{B}$ be a field with $|\mathbb{B}| \geq rn$ and $E = \left\{\lambda_{i,j}\right\}_{i\in[n], j\in[0:r-1]}$ be a set containing rn distinct elements in the field $\mathbb{B}$. We refer to the elements of the set E as evaluation points. For $\ell = (n-k)^n = r^n$, let $\mathcal{C}(E) \subseteq \mathbb{B}^{n\ell}$ denote the linear array code defined by the following $r\ell \times n\ell$ parity check matrix (cf. (3)).

$$\mathbf{H}^{E} = \begin{pmatrix} \mathbf{I} & \mathbf{I} & \cdots & \mathbf{I} \\ A_{1,E} & A_{2,E} & \cdots & A_{n,E} \\ A_{2,E}^2 & A_{2,E}^2 & \cdots & A_{n,E}^2 \\ \vdots & \vdots & \ddots & \vdots \\ A_{1,E}^{r-1} & A_{2,E}^{r-1} & \cdots & A_{n,E}^{r-1} \end{pmatrix} \in \mathbb{B}^{r\ell \times n\ell}, \tag{23}$$

where $\mathbf{I}$ denotes the $\ell \times \ell$ identity matrix. Moreover, for $i \in [n]$, the $\ell \times \ell$ matrix $A_{i,E}$ is defined as follows.

$$A_{i,E} = \sum_{a=0}^{\ell-1} \lambda_{i,a_i} \mathbf{e}_a \mathbf{e}_a^{\mathsf{T}} = \sum_{a=0}^{r^n-1} \lambda_{i,a_i} \mathbf{e}_a \mathbf{e}_a^{\mathsf{T}} \in \mathbb{B}^{\ell\times\ell}, \tag{24}$$

where $\{\mathbf{e}_a\}_{a\in[0:r^n-1]}$ denotes the collection of $\ell = r^n$ standard basis vectors in $\mathbb{B}^\ell$, i.e., all but a-th coordinate of the vector $\mathbf{e}_a$ are equal to 0 and the a-th coordinate has its entry equal to 1.

In [7], Ye and Barg show that the linear array code $\mathcal{C}(E)$ is an MSR code with $t = n - 1$. Note that the sub-packetization level of this MDS code with optimal repair bandwidth is $\ell = r^n$, which is exponential in the code length n. We briefly describe the repair scheme for the code $\mathcal{C}(E)$ in Appendix VII. This linear repair scheme for $\mathcal{C}(E)$ as presented in [7] can be expressed in the form repair matrices (cf. Sec. II-B). For $i \in [n]$, the $\ell \times r\ell$ repair matrix enabling repair of the i-th code block takes the following special block diagonal form with identical $\frac{\ell}{r} \times \ell$ sized diagonal blocks.

$$S_i = \mathrm{Diag}(D_i, D_i, \ldots, D_i) \tag{25}$$

The rows and columns of the $\frac{\ell}{r} \times \ell$ matrix D_i are indexed by the sets $[0 : r^{n-1} - 1]$ and $[0 : r^n - 1]$, respectively. For $b \in [0 : r^{n-1} - 1]$ and $a \in [0 : r^n - 1]$, let $(b_{n-1}, b_{n-2}, \ldots, b_1) \in [0 : r - 1]^{n-1}$ and $(a_n, a_{n-1}, \ldots, a_1) \in$

$[0:r-1]^n$ denote their r-ary vector representations, respectively. With this notaion in place, we have

$$D_i(b,a) = \begin{cases} 1 & \text{if} \quad a_{\setminus\{i\}} = b, \\ 0 & \text{otherwise.} \end{cases} \tag{26}$$

Here, $a_{\setminus\{i\}} = b$, if we have

$$(b_{n-1}, b_{n-2}, \ldots, b_1) = (a_n, a_{n-1}, \ldots, a_{i+1}, a_{i-1}, \ldots, a_1).$$

Note that each of the $\frac{\ell}{r} = r^{n-1}$ rows of the matrix D_i has exactly $r = n-k$ non-zero entries. For $b \in [0 : r^{n-1}-1]$, $a \in [0 : r^n - 1]$ and $w \in [r-1]$, we have that

$$D_i A^w_{i,E}(b,a) = \begin{cases} \lambda^w_{i,a_i} & \text{if} \quad a_{\setminus\{i\}} = b, \\ 0 & \text{otherwise.} \end{cases} \tag{27}$$

and

$$D_i A^w_{j,E}(b,a) = \begin{cases} \lambda^w_{i,b_j} & \text{if} \quad a_{\setminus\{i\}} = b, \\ 0 & \text{otherwise.} \end{cases} \tag{28}$$

Now, for any distinct $w_1, w_2 \in [0 : r-1]$, it is straightforward to verify the following.

$$D_i A^{w_1}_{j,E} \cap D_i A^{w_2}_{j,E} = \begin{cases} \{0\} & \text{if} \;\; i = j, \\ D_i & \text{otherwise,} \end{cases} \tag{29}$$

where we use matrices to denote their row spaces. For the underlying partiy check matrix $\mathbf{H}^E$ (cf. (23)) and repair matrix S_i (cf. (25)), two kind of matrices involved in the linear repair scheme takes the following form (cf. (5) & (6)).

$$\operatorname{rank}\left(S_i \begin{bmatrix} H_{1,i} \\ \vdots \\ H_{r,i} \end{bmatrix} \right) = \operatorname{rank}\left(\begin{bmatrix} D_i \\ D_i A_{i,E} \\ \vdots \\ D_i A^{r-1}_{i,E} \end{bmatrix} \right) \overset{(i)}{=} \ell, \tag{30}$$

and

$$\sum_{j \in [n]\setminus\{i\}} \operatorname{rank}\left(S_i \begin{bmatrix} H_{1,j} \\ \vdots \\ H_{r,j} \end{bmatrix} \right) = \sum_{j \in [n]\setminus\{i\}} \operatorname{rank}\left(\begin{bmatrix} D_i \\ D_i A_{j,E} \\ \vdots \\ D_i A^{r-1}_{j,E} \end{bmatrix} \right) \overset{(ii)}{=} (n-1)\operatorname{rank}(D_i), \tag{31}$$

where the steps (i) and (ii) follow from (29).

B. Using Ye and Barg codes in Construction 1

We now illustrate how one can select a code from a family of MSR codes obtained by the Ye and Barg costruction as the short MSR code $\mathcal{C}^I$ in Construction 1. Let $\mathcal{C}^{II}$ be the $(N, M = q^{NR} = |\mathcal{C}^{II}|, D)$ code that we combine with the MSR code $\mathcal{C}^I$ in Construction 1. Let $\mathbb{B}$ be a finite field such that we have $|\mathbb{B}| \geq |\mathcal{C}^{II}|rn + 1 = q^{NR}rn + 1$ with

a multiplicative sub-group E of order $r n$, i.e., $|E| = r n$. Note that the sub-group has $\frac{|\mathbb{B}^*|}{|E|} \geq q^{NR}$ cosets, each of size $|E| = r n$. Furthermore, each coset of the sub-group E has the following form.

$$T = \sigma \cdot E = \{\sigma \nu \ : \ \nu \in E\}, \tag{32}$$

where $\sigma \in \mathbb{B}^* := \mathbb{B} \backslash \{0\}$ such that $\sigma \notin E$. We associate q^{NR} distinct cosets of the sub-group E to q^{NR} different codewords of the code $\mathcal{C}^{II}$. For a codeword $\mathbf{c} \in \mathcal{C}^{II}$, let $\sigma_{\mathbf{c}} \in \mathbb{B}^*$ be such that the coset associated with the codeword $\mathbf{c}$ is $\sigma_{\mathbf{c}} \cdot E$.

We take $\mathcal{C}^I$ to be the code obtained from the Ye and Barg construction with $r n$ distinct elements of the multiplicative subgroup E forming the $r n$ evaluation points $\{\lambda_{i,j}\}_{i \in [n], j \in [0:r-1]}$. Recall that the code $\mathcal{C}^I$ is defined by the parity check matrix $\mathbf{H}^E$ (cf. (23)), where n thick columns of the parity-check matrix $\mathbf{H}^E$ corrsponding to n distinct code blocks in a codeword of $\mathcal{C}^I$ are defined by the n distinct $\ell \times \ell$ matrices $\{A_{1,E}, A_{2,E}, \ldots, A_{n,E}\}$. In order to fully specify the code $\mathcal{C}$ obtained from Construciton 1, we also need to specify the scalar $\{\alpha_{j,\mathbf{c}}\}_{j \in [r], \mathbf{c} \in \mathcal{C}^{II}}$ (cf. (11)). For $j \in [r]$ and $\mathbf{c} \in \mathcal{C}^{II}$, we assign

$$\alpha_{j,\mathbf{c}} = \sigma_{\mathbf{c}}^{j-1},$$

where, as defined earlier, $\sigma_{\mathbf{c}}$ specifies the coset of E which is associated with the codeword $\mathbf{c} \in \mathcal{C}^{II}$.

Let $\mathcal{H}$ be the $rN\ell \times MN\ell$ parity check matrix of the code $\mathcal{C}$ obtained from Construction 1. Recall that a codeword of $\mathcal{C}$ has M code blocks which are indexed by the codewords of $\mathcal{C}^{II}$. Given the aforementioned choice for the short MSR code $\mathcal{C}^I$, the $N\ell$ columns of $\mathcal{H}$ corresponding to the code block indexed by $\mathbf{c} \in \mathcal{C}^{II}$ takes the following form (cf. (11)).

$$\mathcal{H}_{\mathbf{c}} = \begin{pmatrix} \mathrm{Diag}(\mathbf{I}, \mathbf{I}, \ldots, \mathbf{I}) \\ \sigma_{\mathbf{c}} \cdot \mathrm{Diag}(A_{c_1,E}, A_{c_2,E}, \ldots, A_{c_N,E}) \\ \sigma_{\mathbf{c}}^2 \cdot \mathrm{Diag}(A_{c_1,E}^2, A_{c_2,E}^2, \ldots, A_{c_N,E}^2) \\ \vdots \\ \sigma_{\mathbf{c}}^{r-1} \cdot \mathrm{Diag}(A_{c_1,E}^{r-1}, A_{c_2,E}^{r-1}, \ldots, A_{c_N,E}^{r-1}) \end{pmatrix} = \begin{pmatrix} \mathbf{I} \\ \sigma_{\mathbf{c}} \cdot \mathcal{A}_{\mathbf{c},E} \\ \sigma_{\mathbf{c}}^2 \cdot \mathcal{A}_{\mathbf{c},E}^2 \\ \vdots \\ \sigma_{\mathbf{c}}^{r-1} \cdot \mathcal{A}_{\mathbf{c},E}^{r-1} \end{pmatrix}, \tag{33}$$

where $\mathbf{I}$ denotes both $\ell \times \ell$ and $N\ell \times N\ell$ identity matrices. Moreover, we use $\mathcal{A}_{\mathbf{c},E}$ to denote the following $N\ell \times N\ell$ block diagonal matrix

$$\mathrm{Diag}(A_{c_1,E}, A_{c_2,E}, \ldots, A_{c_N,E}). \tag{34}$$

1) Repair bandwidth for repairing a single code block (node) $\mathcal{C}$: As show in the proof of Theorem III.1, the code block indexed by $\mathbf{c} \in \mathcal{C}^{II}$ in a codeword of $\mathcal{C}$ can be repaired using the following $N\ell \times rN\ell$ repair matrix.

$$\mathcal{S}_{\mathbf{c}} = \mathrm{Diag}(S_{c_1}, S_{c_2}, \ldots, S_{c_N}), \tag{35}$$

where $\ell \times r\ell$ matrices $\{S_{c_i}\}_{i \in [N]}$ are defined in (25). Taking the code $\mathcal{C}^{II}$ with large enough distance, the repair bandwidth associated with linear repair scheme defined by these repair matrices can be made at most $(1+\epsilon) \cdot \frac{N\ell}{r}$ (cf. Remark 2).

2) MDS property of $\mathcal{C}$: Next, we argue that the code $\mathcal{C}$ obtained in this section is an MDS code. Along with the previous result on its repair bandwidth, the following result establishes that $\mathcal{C}$ is an ϵ-MSR code.

Lemma IV.1. *Let $\mathcal{C}$ be a linear array code defined by the $rN\ell \times q^{NR}N\ell$ parity check matrix $\mathcal{H}$ as described in* (33)*. Then, $\mathcal{C}$ is a $[q^{NR}, q^{NR}-r, r+1, N\ell]_{\mathbb{B}}$ MDS code.*

Proof. In order to argue that $\mathcal{C}^{\text{new}}$ is an MDS code, we need to show that any $rN\ell \times rN\ell$ sub-matrix of $\mathcal{H}$ consisting of $r = n-k$ thick columns of $\mathcal{H}$ corresponding to any r distinct code blocks is full rank. Let's consider the r code blocks indexed by the following r codewords of $\mathcal{C}^{\text{II}}$.

$$\mathcal{R} = \left\{\mathbf{c}^1, \mathbf{c}^2, \ldots, \mathbf{c}^r\right\} \subset \mathcal{C}^{\text{II}}.$$

The $rN\ell \times rN\ell$ sub-matrix of $\mathcal{H}$ that corresponds to the code blocks indexed by these codewords takes the following form.

$$\begin{aligned}\mathcal{H}_{\mathcal{R}} &= \begin{pmatrix} \mathcal{H}_{\mathbf{c}^1} & \mathcal{H}_{\mathbf{c}^2} & \cdots & \mathcal{H}_{\mathbf{c}^r} \end{pmatrix} \\ &= \begin{pmatrix} \mathbf{I} & \mathbf{I} & \cdots & \mathbf{I} \\ \sigma_{\mathbf{c}^1}\cdot\mathcal{A}_{\mathbf{c}^1,E} & \sigma_{\mathbf{c}^2}\cdot\mathcal{A}_{\mathbf{c}^2,E} & \cdots & \sigma_{\mathbf{c}^r}\cdot\mathcal{A}_{\mathbf{c}^r,E} \\ \sigma^2_{\mathbf{c}^1}\cdot\mathcal{A}_{\mathbf{c}^1,E} & \sigma^2_{\mathbf{c}^2}\cdot\mathcal{A}^2_{\mathbf{c}^2,E} & \cdots & \sigma^2_{\mathbf{c}^r}\cdot\mathcal{A}^2_{\mathbf{c}^r,E} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma^{r-1}_{\mathbf{c}^1}\cdot\mathcal{A}_{\mathbf{c}^1,E} & \sigma^{r-1}_{\mathbf{c}^2}\cdot\mathcal{A}^{r-1}_{\mathbf{c}^2,E} & \cdots & \sigma^{r-1}_{\mathbf{c}^r}\cdot\mathcal{A}^{r-1}_{\mathbf{c}^r,E} \end{pmatrix}. \end{aligned} \tag{36}$$

Taking the block diagonal sturucture of the matrices $\{\mathcal{A}_{\mathbf{c}^w,E}\}_{w\in[r]}$ into account (cf. (34)), it is suffienct to argue that for every $i \in [N]$ the following matrix is full rank.

$$\mathbf{U}_{\mathcal{R},i} = \begin{pmatrix} \mathbf{I} & \mathbf{I} & \cdots & \mathbf{I} \\ \sigma_{\mathbf{c}^1}\cdot A_{c^1_i,E} & \sigma_{\mathbf{c}^2}\cdot A_{c^2_i,E} & \cdots & \sigma_{\mathbf{c}^r}\cdot A_{c^r_i,E} \\ \sigma^2_{\mathbf{c}^1}\cdot A_{c^1_i,E} & \sigma^2_{\mathbf{c}^2}\cdot A^2_{c^2_i,E} & \cdots & \sigma^2_{\mathbf{c}^r}\cdot A^2_{c^r_i,E} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma^{r-1}_{\mathbf{c}^1}\cdot A_{c^1_i,E} & \sigma^{r-1}_{\mathbf{c}^2}\cdot A^{r-1}_{c^2_i,E} & \cdots & \sigma^{r-1}_{\mathbf{c}^r}\cdot A^{r-1}_{c^r_i,E} \end{pmatrix}], \tag{37}$$

where c^j_i denotes the i-th code symbol in the codeword $\mathbf{c}^j \in \mathcal{R} \subset \mathcal{C}^{\text{II}}$. For any $i \in [n]$, $\mathbf{U}_{\mathcal{R},i}$ is a block matrix with diagonal blocks (cf. (24)). Similar to the proof of Theorem III.2 in [7], one can rearrange the rows and columns of the matrix $\mathbf{U}_{\mathcal{R},i}$ to obtain a block *diagonal* matrix, where diagonal blocks are Vandermode matrices. Therefore, the matrix $\mathbf{U}_{\mathcal{R},i}$ is a full rank matrix. This completes the proof. □

We highlight the relationship between the length and subpacketization level of the obtained family of ϵ-MSR codes in the following result.

Theorem IV.1. *Given an $\epsilon > 0$, there exists a constant $s = s(\epsilon) > 0$ such that for infinite values of l there exists an $(\mathcal{N} = \exp(sl), \mathcal{K} = \exp(sl) - r, \mathcal{T} = \mathcal{N} - 1, l)_{\mathbb{B}}$ ϵ-MSR code. Furthermore, the required field size $|\mathbb{B}|$ scales as $O(\mathcal{N})$.*

Proof. Recall that it follows from GV bound that for every alphabet of size q and $\delta \in (0, 1/q]$, there exists a code over the alphabet with relative minimum distance at least δ and rate

$$R \geq 1 - h_q(\delta) - o(1), \tag{38}$$

where $h_q(x) = x\log_q(q-1) - x\log_q x - (1-x)\log_q(1-x)$ denotes the q-ary entropy function. For a constant $\epsilon > 0$, we choose q such that $\delta^* = 1 - \frac{\epsilon}{r-1} < 1 - 1/q$. Now we take $\mathcal{C}^{II}$ to be an N-length code over an alphabet of size q such that the code has $q^{N(1-h_q(\delta^*)-o(1))}$ codewords and relative minimum distance at least $\delta^* = 1 - \frac{\epsilon}{r-1}$ (cf. (38)). We combine this with the an $(n = q, k = q-r, t = q-1, \ell = r^q)_{\mathbb{B}}$ MSR code from [7] as described above. This gives us an ϵ-MSR code with length $\mathcal{N} = q^{N(1-h_q(\delta^*)-o(1))}$ and sub-packetization level $l = N\ell = Nr^q$. Therefore, we have

$$\mathcal{N} = q^{((1-h_q(\delta^*)-o(1))/r^q)l}. \tag{39}$$

For constant r and q, this can be expressed as $\mathcal{N} = \exp(sl)$ for a suitable constant s. Note that for constant r and q, the required filed size $q^{N(1-h_q(\delta^*)-o(1))}qr + 1$ scales as $O(\mathcal{N})$. □

Remark 4. Note that the sub-packetization level of the ϵ-MSR codes mentioned in Theorem IV.1 satisfy $l = O\left(r^{(r/\epsilon)} \cdot \log \mathcal{N}\right)$. For the identical repair-bandwidth gurantees, the MDS codes obtained in [15] require a smaller sub-packetization level of $r^{1/\epsilon}$. However, as compared to [15], the codes mentioned in Theorem IV.1 ensure load balancing during the repair process and require field size which is only linear in the code length $\mathcal{N}$ when $r = \Theta(1)$. We note that the codes constructed in [15] require field size which is exponential in the code length.

V. Necessary Sub-packetization for ϵ-MSR Codes

In Sec. IV, we establish that resorting to ϵ-MSR codes for positive ϵ allows the number of nodes to scale exponentially with the sub-packetization level ℓ. This is an encouraging result, since for optimal MSR codes with constant $r = n-k$, it is known that n has to scale polylogarithmically with ℓ [11], [17]. In this section we derive an upper bounds on the number of nodes in an ϵ-MSR codes. The bound rely on similar technique that are employed in [11], [17] to bound the number of nodes in an MSR code. Each node is assigned a vector in some vector space. Then it is shown that the assigned vectors are linearly independent and hence the number of such vectors (and nodes) is at most the dimension of the vector space.

Theorem V.1. *In an $(n, k, t = n-1, \ell)$ ϵ−MSR code, n the number of nodes in the system is upper bounded by $\ell^{\frac{\ell}{r}(1+\epsilon)+1}$.*

Proof: The proof is similar to the proof of Theorem 4 in [17]. Let the parity check matrix of the code be the $r\ell \times n\ell$ matrix

$$\mathbf{H} = \begin{pmatrix} \mathbf{H}_1 & \cdots & \mathbf{H}_n \end{pmatrix}, \tag{40}$$

where each $\mathbf{H}_i$ is an $r\ell \times \ell$ matrix. By the ϵ-recoverability of node i, there exists an $\ell \times r\ell$ matrix S_i which satisfies

$$\text{rank}(S_i\mathbf{H}_j) = \begin{cases} \ell & i = j \\ \frac{\ell}{r}(1+\epsilon) & \text{else.} \end{cases} \tag{41}$$

Since the $\ell \times \ell$ matrix $S_i \mathbf{H}_i$ is of full rank there exist a subset of columns C_i of size $u = \frac{\ell}{r}(1+\epsilon)+1$, such that the restriction of $S_i \mathbf{H}_i$ to the columns in C_i and the first u rows, is a matrix of full rank, i.e.,

$$\text{rank}(S_i \mathbf{H}_i)_{[u],C_i} = u.$$

Define for node i the polynomial $f_i : \mathbb{F}^{\ell \times r\ell} \to \mathbb{F}$ as

$$f_i(X) = \det(X\mathbf{H}_i)_{[u],C_i},$$

where $X = (x_{i,j})$ is an $\ell \times r\ell$ matrix in the variables $x_{i,j}$. By (41) it follows that

$$f_i(S_j) = \begin{cases} \neq 0 & i = j \\ 0 & i \neq j, \end{cases}$$

and therefore the n polynomials f_i are linearly independent. If we assume the contrary, then there exists scalars α_i not all zeros such that $\sum_j \alpha_j f_j = 0$. However by plugging S_i on both sides of the equation we get that

$$0 = \sum_j \alpha_j f_j(S_j) = \alpha_i f_i(S_i).$$

Since $f_i(S_i) \neq 0$ we get that $\alpha_i = 0$. By repeating this argument for all i's, we get to a contradiction.

Each polynomial f_i is a homogenous polynomial of degree u, spanned by the monomials of the form $\prod_{m=1}^{u} x_{m,j_m}$. Clearly there are ℓ^u such monomials, and therefore the polynomials f_i form a set of linearly independent vectors in a vector space of dimension ℓ^u. The result follows since the size of such set can not exceed the dimension. ■

VI. Conclusion

We present a general approach to construct ϵ-MSR codes – exact-repairable MDS code with small sub-packetization level and near-optimal repair bandwidth for the repair of a single code block. The obtained codes using the proposed approach also ensure load balancing among the contacted code blocks during the repair process. The proposed construction requires contacting all the intact code blocks for the regeneration of a failed code block. Extending these results to the settings which require contacting smaller number of intact code blocks and/or demand simultaneous repair of multiple code blocks is an interesting and immediate direction to pursue. We also present a lower bound on the sub-packetization level that is necessary for an ϵ-MSR code. However, there is a gap between this bound and the sub-packetization level achieved by our constructions. The efforts to bridge this gap are part of our ongoing work.

References

[1] A. G. Dimakis, K. Ramchandran, Y. Wu, and C. Suh. A survey on network codes for distributed storage. *Proc. of the IEEE*, 99(3):476–489, 2011.

[2] A. G. 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.

[3] W. Huang, M. Langberg, J. Kliewer, and J. Bruck. Communication efficient secret sharing. *CoRR*, abs/1505.07515, 2015.

[4] 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.

[5] I. Tamo, Z. Wang, and J. Bruck. Zigzag codes: MDS array codes with optimal rebuilding. *IEEE Trans. Inf. Theory*, 59(3):1597–1616, 2013.

[6] D. S. Papailiopoulos, A. G. Dimakis, and V. Cadambe. Repair optimal erasure codes through hadamard designs. *IEEE Trans. Inf. Theory*, 59(5):3021–3037, 2013.

[7] M. Ye and A. Barg. Explicit constructions of high-rate MDS array codes with optimal repair bandwidth. *CoRR*, abs/1604.00454, 2016.

[8] M. Ye and A. Barg. Explicit constructions of optimal-access MDS codes with nearly optimal sub-packetization. *CoRR*, abs/1605.08630, 2016.

[9] B. Sasidharan, M. Vajha, and P. V. Kumar. An explicit, coupled-layer construction of a high-rate MSR code with low sub-packetization level, small field size and all-node repair. *CoRR*, abs/1607.07335, 2016.

[10] A. S. Rawat, O. O. Koyluoglu, and S. Vishwanath. Progress on high-rate MSR codes: Enabling arbitrary number of helper nodes. *CoRR*, abs/1601.06362, 2016.

[11] S. Goparaju, I. Tamo, and R. Calderbank. An improved sub-packetization bound for minimum storage regenerating codes. *IEEE Trans. Inf. Theory*, 60(5):2770–2779, May 2014.

[12] O. Khan, R. Burns, J. Plank, W. Pierce, and C. Huang. Rethinking erasure codes for cloud file systems: Minimizing I/O for recovery and degraded reads. In *Proc. of 10th USENIX Conference on File and Storage Technologies (FAST)*, 2012.

[13] K. V. Rashmi, N. B. Shah, and K. Ramchandran. A piggybacking design framework for read-and download-efficient distributed storage codes. In *Proc. of 2013 IEEE International Symposium on Information Theory (ISIT)*, pages 331–335, July 2013.

[14] I. Tamo and K. Efremenko. New results on MSR codes. In *Information Theory and Applications Workshop (ITA), 2016*, Feb 2016.

[15] V. Guruswami and A. S. Rawat. MDS code constructions with small sub-packetization and near-optimal repair bandwidth. In *Proc. of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2017.

[16] Z. Wang, I. Tamo, and J. Bruck. Explicit minimum storage regenerating codes. *IEEE Trans. Inf. Theory*, 62(8):4466–4480, Aug 2016.

[17] I. Tamo, Z. Wang, and J. Bruck. Access versus bandwidth in codes for storage. *IEEE Trans. Inf. Theory*, 60(4):2028–2037, April 2014.

VII. Single Node Repair in Ye and Barg Construction [7]

Let the i-th code block $\mathbf{c}_i = \left(c_{i,0}, c_{i,1}, \ldots, c_{i,\ell-1}\right)$ be the code block being repaired. Note that all the block in the parity check matrix $\mathbf{H}^E$ (cf. (23)) are diagonal matrices (cf. (24)). Therefore, the partiy check constraints defining the code $\mathcal{C}(E)$ can be rewritten as follows.

$$\sum_{i=1}^{n} \lambda_{i,a_i}^{w} c_{i,a} = 0 \qquad \forall \;\; w = [0 : n-k-1] \text{ and } a = [0 : \ell - 1]. \tag{42}$$

The repair mechanism recovers the $r = n - k$ symbols

$$\left\{c_{i,a(i,0)}, c_{i,a(i,1)}, \ldots, c_{i,a(i,r-1)}\right\}$$

with the help of the following set of symbols downloaded from the remaining $t = n - 1$ code blocks.

$$\mu_{j,i}^{(a)} := \sum_{u=0}^{r-1} c_{j,a(i,u)}, \qquad j \in [n] \backslash \{i\}. \tag{43}$$

In particular, for $a \in \{0, 1, \ldots, \ell - 1\}$ and $u \in \{0, 1, \ldots, r-1\}$, it follows from (42) that

$$\lambda_{i,u}^{w} c_{i,a(i,u)} + \sum_{j \neq i} \lambda_{j,a_j}^{w} c_{j,a(i,u)} = 0. \tag{44}$$

Summing (44) over $u = 0, 1, \ldots, r-1$, we obtain the following system of equations.

$$\begin{pmatrix} 1 & 1 & \cdots & 1 \\ \lambda_{i,0} & \lambda_{i,1} & \cdots & \lambda_{i,r-1} \\ \lambda_{i,0}^2 & \lambda_{i,1}^2 & \cdots & \lambda_{i,r-1}^2 \\ \vdots & \vdots & \ddots & \vdots \\ \lambda_{i,0}^{r-1} & \lambda_{i,1}^{r-1} & \cdots & \lambda_{i,r-1}^{r-1} \end{pmatrix} \begin{pmatrix} c_{i,a(i,0)} \\ c_{i,a(i,1)} \\ c_{i,a(i,2)} \\ \vdots \\ c_{i,a(i,r-1)} \end{pmatrix} = - \begin{pmatrix} \sum_{u=0}^{r-1} \sum_{j \neq i} c_{j,a(i,u)} \\ \sum_{u=0}^{r-1} \sum_{j \neq i} \lambda_{j,a_j} c_{j,a(i,u)} \\ \sum_{u=0}^{r-1} \sum_{j \neq i} \lambda_{j,a_j}^2 c_{j,a(i,u)} \\ \vdots \\ \sum_{u=0}^{r-1} \sum_{j \neq i} \lambda_{j,a_j}^{r-1} c_{j,a(i,u)} \end{pmatrix},$$

or

$$\begin{pmatrix} 1 & 1 & \cdots & 1 \\ \lambda_{i,0} & \lambda_{i,1} & \cdots & \lambda_{i,r-1} \\ \lambda_{i,0}^2 & \lambda_{i,1}^2 & \cdots & \lambda_{i,r-1}^2 \\ \vdots & \vdots & \ddots & \vdots \\ \lambda_{i,0}^{r-1} & \lambda_{i,1}^{r-1} & \cdots & \lambda_{i,r-1}^{r-1} \end{pmatrix} \begin{pmatrix} c_{i,a(i,0)} \\ c_{i,a(i,1)} \\ c_{i,a(i,2)} \\ \vdots \\ c_{i,a(i,r-1)} \end{pmatrix} = - \begin{pmatrix} \sum_{j \neq i} \mu_{j,i}^{(a)} \\ \sum_{j \neq i} \lambda_{j,a_j} \mu_{j,i}^{(a)} \\ \sum_{j \neq i} \lambda_{j,a_j}^2 \mu_{j,i}^{(a)} \\ \vdots \\ \sum_{j \neq i} \lambda_{j,a_j}^{r-1} \mu_{j,i}^{(a)} \end{pmatrix}. \tag{45}$$

Since $\{\lambda_{i,0}, \lambda_{i,1}, \ldots, \lambda_{i,r-1}\}$ are all distinct elements, this system of equations can be solved for the desired code symbols $\left\{c_{i,a(i,0)}, c_{i,a(i,1)}, \ldots, c_{i,a(i,r-1)}\right\}$.