

# **On Epsilon-MSCR Codes for Two Erasures**

by

Rekha Devi Bhoopatiraju, Lalitha Vadlamani

in

*The 2019 IEEE International Symposium on Information Theory (ISIT)*

Maison de la Mutualité in Paris

Report No: IIIT/TR/2019/-1



Centre for Communications  
International Institute of Information Technology  
Hyderabad - 500 032, INDIA  
July 2019

# On Epsilon-MSCR Codes for Two Erasures

Bh. Rekha Devi, V. Lalitha

SPCRC, International Institute of Information Technology, Hyderabad, India

Email: rekha.devi@research.iiit.ac.in, lalitha.v@iiit.ac.in

## Abstract

Cooperative regenerating codes are regenerating codes designed to tradeoff storage for repair bandwidth in case of multiple node failures. Minimum storage cooperative regenerating (MSCR) codes are a class of cooperative regenerating codes which achieve the minimum storage point of the tradeoff. Recently, these codes have been constructed for all possible parameters  $(n, k, d, h)$ , where  $h$  erasures are repaired by contacting any  $d$  surviving nodes. However, these constructions have very large sub-packetization.  $\epsilon$ -MSR codes are a class of codes introduced to tradeoff subpacketization level for a slight increase in the repair bandwidth for the case of single node failures. We introduce the framework of  $\epsilon$ -MSCR codes which allow for a similar tradeoff for the case of multiple node failures. We present a construction of  $\epsilon$ -MSCR codes, which can recover from two node failures, by concatenating a class of MSCR codes and scalar linear codes. We give a repair procedure to repair the  $\epsilon$ -MSCR codes in the event of two node failures and calculate the repair bandwidth for the same. We characterize the increase in repair bandwidth incurred by the method in comparison with the optimal repair bandwidth given by the cut-set bound. Finally, we show the subpacketization level of  $\epsilon$ -MSCR codes scales logarithmically in the number of nodes.

## I. INTRODUCTION

In an  $(n, k, d, l)$  regenerating code [1], a file comprised of  $B$  symbols from a finite field  $\mathbb{F}_q$  is encoded into a set of  $nl$  code symbols and they are stored across  $n$  nodes in the network with each node storing  $l$  code symbols. The parameter  $l$  is called the sub-packetization level of the code. A data collector can download the data by connecting to any  $k$  nodes. In the event of node failure, node repair is carried out by having the replacement node connect to any  $d$  nodes and download  $\beta \leq l$  symbols from each node. A cut-set bound on the number of symbols downloaded from each of the  $d$  nodes for the repair of a single node was given in [1] as

$$RB \geq \frac{l}{d - k + 1}. \quad (1)$$

The codes which achieve the above cut-set bound with equality are termed as minimum storage regenerating (MSR) codes. Explicit constructions of MSR codes for  $d \geq 2k - 2$  are given in [2] using the product-matrix framework. Employing Hadamard designs, MSR codes which achieve high-rate for two parity nodes were presented in [3]. In [4], existence of MSR codes was shown for all parameters  $(n, k, d)$  as  $l \rightarrow \infty$ . Explicit construction of zigzag codes (high-rate systematic repair MSR codes) were given in [5] for  $d = n - 1$ . However, these codes require a sub-packetization that is exponential in  $k$ . Explicit construction of MSR codes for all parameters  $(n, k, d)$  with  $l = (d - k + 1)^n$  were given in [6]. The sub-packetization level  $l$  was improved to  $r^{\frac{n}{r}}$  ( $r = n - k$ ) for  $d = n - 1$  by the same authors in [7]. With the help of coupled-layer construction, MSR code with parameters  $(n = qt, k = q(t - 1), d = n - 1, l = q^t = r^{\frac{n}{r}})$  was presented in [8] for auxiliary parameters  $q \geq 2, t \geq 2$ . A lower bound on sub-packetization level  $l$  for fixed  $k$  and  $n$  has been discussed in [9].

### A. $\epsilon$ -MSR Codes

These are codes which tradeoff subpacketization for slight increase in repair bandwidth and are obtained by the concatenation of a MSR code with a scalar linear code having large minimum distance. For the repair of any failed node, the amount of information downloaded from each helper node is at-most  $(1 + \epsilon)$  times that of the optimal for some  $\epsilon > 0$ . For constant  $r = n - k$ , the subpacketization level  $l$  scales logarithmically with the number of nodes.  $\epsilon$ -MSR codes are known for two cases (i) All the remaining nodes are contacted for repairing the failed node [10]. (ii) Only a subset of the remaining nodes are contacted for repair [11]. In this case, a failed node can be repaired by contacting a set of  $P'$  compulsory nodes and  $P - P'$  arbitrary nodes.

### B. Repair of Multiple Erasures

**Cooperative Repair:** In cooperative repair for  $h$  node failures, node repair is carried out in two rounds. In the first round, each of the replacement node connect to any  $d$  nodes of the  $n - h$  surviving nodes and download  $\beta_1 \leq \alpha$  symbols from each node. In the second round, the replacement nodes exchange data among themselves. Every replacement node downloads  $\beta_2$  symbols from every other replacement node. Hence, the repair bandwidth per replacement node is given by

$$\gamma = d\beta_1 + (h - 1)\beta_2.$$

The cut-set bound on repair bandwidth for  $h$  failed nodes under cooperative repair ([12]) is given by

$$RB \geq \frac{h(h+d-1)l}{h+d-k}. \quad (2)$$

The codes which achieve the above cut-set bound with equality are termed as minimum storage cooperative regenerating (MSCR) codes. Constructions of MSCR codes for  $d = k$  were proposed in [13]. Constructions of MSCR codes for  $d \geq 2k - 3$  which can repair two erasures were presented in [14]. For general values of  $h, d$ , MSCR codes were given in [15] which achieve optimal repair bandwidth. These codes require large sub-packetization level  $l \approx (d - k + 1)^{h \binom{n}{h}}$ . In a recent work, [16], explicit constructions of scalar MSCR codes ( $\beta = 1$ ) for  $d \geq \max\{2k - 1 - h, k\}$  and  $l = d - k + h$  were constructed using invariant repair spaces.

**Centralized Repair:** In centralized repair of  $h$  node failures, a central node recovers the contents of all the failed nodes by contacting a set of  $d$  helper nodes and downloading  $\beta$  symbols from each of them. The cut-set bound on repair bandwidth for  $h$  failed nodes under centralized repair ([4]) is given by

$$RB \geq \frac{hdl}{h+d-k}. \quad (3)$$

The codes which achieve the above cut-set bound with equality are termed as minimum storage multi-repair (MSMR) codes. The constructions of MSMR codes based on product-matrix codes and interference-alignment based codes have been investigated in [17].

### C. Our Contributions

In this paper, we consider the problem of constructing near-optimal minimum storage cooperative regenerating codes for two erasures.

- We introduce  $\epsilon$ -MSCR code framework.
- We give a construction of  $\epsilon$ -MSCR codes which are obtained by concatenating an MSCR code (which can repair two erasures) and a scalar linear code. A quick review of MSCR codes with specific parameters is given in Section II and the construction of  $\epsilon$ -MSCR codes itself is presented in Section III.
- We present a method to repair the code under two erasures (Section IV) and calculate the repair bandwidth incurred by the method (Section V).
- We characterize the  $\epsilon$  resulting from the repair method under consideration (Section V). It turns out that  $\epsilon$  is a finite value and cannot be made arbitrarily small.
- We also show that these codes have sub-packetization level logarithmic in the number of nodes (Section V).

## II. REVIEW OF MSCR CODES

In this section, we will introduce vector MDS codes and note that MSCR codes are vector MDS codes. Subsequently, we will present an  $(n, k, d = k + 1, h = 2, l = 3^{\binom{n}{2}})$  MSCR code which can recover from 2 erasures by contacting  $k + 1$  nodes.

**Vector MDS Codes:** A linear code  $\mathcal{C}_{vec}$  is said to be a vector code with parameters  $(n, K_S, d_{\min}, l)$  if  $nl$  code symbols are grouped into  $n$  vector-code symbols and each vector-code symbol is of size  $l$  over  $\mathbb{B}$ . The minimum distance  $d_{\min}$  of  $\mathcal{C}_{vec}$  is the minimum number of vector-code symbols in which any two codewords differ or equivalently minimum weight of any non-zero codeword. The  $nl$  code symbols themselves form a  $[nl, K_S]$  scalar code of dimension  $K_S$  over  $\mathbb{B}$ . A vector code can be described in terms of its parity check matrix  $H = [H_1, H_2, \dots, H_n]$  as follows:

$$\sum_{i=1}^n H_i \mathbf{c}^i = 0, \quad (4)$$

where  $H_i, 1 \leq i \leq n$  is a matrix of size  $(nl - K_S) \times l$  and  $\mathbf{c}^i$  is the vector code symbol corresponding to node  $i$ . A vector code is said to be MDS code if  $l \mid K_S$  and  $d_{\min} = n - \frac{K_S}{l} + 1$ . Note that MSCR codes are vector MDS codes.

**MSCR Codes:** We describe a  $(n, k, d = k + 1, h = 2, l = 3^{\binom{n}{2}})$  MSCR code  $\mathcal{C}^Y$  over a finite field  $\mathbb{B}$  via its parity check matrix as follows:

$$H = \begin{bmatrix} I & I & \dots & I \\ H_1 & H_2 & \dots & H_n \\ \vdots & \vdots & \vdots & \vdots \\ H_1^{r-1} & H_2^{r-1} & \dots & H_n^{r-1} \end{bmatrix}, \quad (5)$$

where  $H_i$  is an  $l \times l$  matrix,  $r = n - k$ . To define  $H_i$ , we need to consider the  $l$  scalar code-symbols in a vector code-symbol as indexed by an  $m = \binom{n}{2}$ -length 3-ary representation  $(b_m, b_{m-1}, \dots, b_1)$  and  $b = b_1 + 3b_2 + \dots + b_m 3^{m-1}$ .  $H_i$  is a diagonal matrix, whose  $(b + 1)^{\text{th}}$  diagonal entry is given by  $\lambda_{i,f(i,b)} \in \mathbb{B}$ . In the following, we define the function  $f$ .

- Consider a function  $g(i_1, i_2) = \binom{i_2-1}{2} + i_1, 1 \leq i_1 < i_2 \leq n$ .  $g$  maps every pair  $(i_1, i_2)$  to a unique number in  $\{1, 2, \dots, \binom{n}{2}\}$ .
- Let  $P_f = |\{j : (1 \leq j \leq i-1) \& (b_{g(j,i)} = 2)\}| + |\{j : (i+1 \leq j \leq n) \& (b_{g(i,j)} = 1)\}|$ .

$$f(i, b) = \begin{cases} 0, & \text{if } P_f \text{ is even} \\ 1, & \text{if } P_f \text{ is odd.} \end{cases} \quad (6)$$

It is required that  $\{\lambda_{1,0}, \lambda_{1,1}, \lambda_{2,0}, \lambda_{2,1}, \dots, \lambda_{n,0}, \lambda_{n,1}\}$  are all distinct elements in  $\mathbb{B}$ . The following claim gives the structure of the function  $f$  which will be used in the later parts of the paper.

**Claim 1.** Consider a pair  $(i_1, i_2)$  such that  $1 \leq i_1 < i_2 \leq n$ . We denote  $g(i_1, i_2) = g_{12}$ . Also, let  $b(i, u)$  be defined as follows:

$$b(i, u) = (b_m, b_{m-1}, \dots, b_{i+1}, u, b_{i-1}, \dots, b_1). \quad (7)$$

Then, based on the definition of function  $f$  above, we have

$$\begin{aligned} f(i, b(g_{12}, 0)) &= f(i, b(g_{12}, 1)) = f(i, b(g_{12}, 2)), i \notin \{i_1, i_2\} \\ f(i_1, b(g_{12}, 0)) &= f(i_1, b(g_{12}, 2)) \neq f(i_1, b(g_{12}, 1)), \\ f(i_2, b(g_{12}, 0)) &= f(i_2, b(g_{12}, 1)) \neq f(i_2, b(g_{12}, 2)). \end{aligned}$$

### III. $\epsilon$ -MSCR CODES FOR TWO ERASURES

**Definition 1.** Consider a vector MDS code  $\mathcal{C}$  with parameters  $(n, kl, d_{\min} = n - k + 1, l)$ . For any  $\epsilon > 0$ , we say that  $\mathcal{C}$  is an  $\epsilon$ -MSCR code for  $h$  erasures, if any  $h$  erasures can be repaired by contacting  $d$  of the remaining nodes with a total repair bandwidth which is at most  $(1 + \epsilon) \frac{h(h+d-1)l}{(h+d-k)}$ .

In the following, we give a construction of  $\epsilon$ -MSCR code which can be recovered from two erasures.

**Construction III.1.** An  $\epsilon$ -MSCR code is a vector MDS code obtained by concatenating an MSCR code which can recover from two node failures and a scalar linear code. We will first describe the parameters of the constituent codes and then give the method of concatenation.

**MSCR Code:** The first constituent code is an MSCR code with parameters  $(n, k, d = k + 1, h = 2, l = 3 \binom{n}{2})$  described in Section II. We consider the case when  $r = n - k \geq 5$ .  $\{\lambda_{1,0}, \lambda_{1,1}, \lambda_{2,0}, \lambda_{2,1}, \dots, \lambda_{n,0}, \lambda_{n,1}\} \in \mathbb{B}_0 \subset \mathbb{B}$ , where  $\mathbb{B}_0$  is a multiplicative subgroup of  $\mathbb{B} \setminus \{0\}$ .

**Scalar Code:** The second constituent code is a scalar linear code over  $\mathbb{F}_q, r = n - k < q \leq n$  with length  $N$ , dimension  $K$  and minimum distance  $D = \delta N, 0 < \delta < 1$ . The number of codewords in the scalar code is assumed to be  $M$ . We will denote this code by  $\mathcal{C}^S$ .

**$\epsilon$ -MSCR Code:** Given the above two codes,  $\epsilon$ -MSCR code  $\mathcal{C}$  is a vector MDS code with the parameters  $(M, K_S = (M - r)Nl, d_{\min} = r + 1, L = Nl)$ . The number of nodes in the  $\epsilon$ -MSCR code equals the number of codewords in the scalar code  $\mathcal{C}^S$ . The nodes themselves are indexed by the codewords of the scalar code  $\mathcal{C}^S$ . Let  $\mathbf{a}_i = (a_{i,1}, a_{i,2}, \dots, a_{i,N})$  denote the  $i^{\text{th}}$  codeword of the scalar code. The parity check equation satisfied by the  $\epsilon$ -MSCR code  $\mathcal{C}$  is given by

$$\sum_{i=1}^M \mathcal{H}_i \mathbf{c}^i = 0, \text{ where} \quad (8)$$

$$\mathcal{H}_i = \begin{bmatrix} \text{Diag}(I, I, \dots, I) \\ \sigma_i \text{Diag}(H_{a_{i,1}}, H_{a_{i,2}}, \dots, H_{a_{i,N}}) \\ \vdots \\ \sigma_i^{r-1} \text{Diag}(H_{a_{i,1}}^{r-1}, H_{a_{i,2}}^{r-1}, \dots, H_{a_{i,N}}^{r-1}) \end{bmatrix}. \quad (9)$$

$\{\sigma_i, 1 \leq i \leq M\}$  are picked such that each  $\sigma_i$  belongs to a distinct coset of  $\mathbb{B}_0$  in  $\mathbb{B} \setminus \{0\}$ .

**Theorem III.2.** The code  $\mathcal{C}$  given in Construction III.1 is a vector MDS code.

*Proof:* The code is defined by its  $rNl \times MNl$  parity check matrix  $\mathcal{H}$ . Any  $rNl \times rNl$  sub-matrix of  $\mathcal{H}$  should be full-rank for  $\epsilon$ -MSCR to satisfy the MDS property. Each thick column ( $Nl \times 1$ ) of  $\mathcal{H}$  is indexed by a codeword of  $\mathcal{C}^S$  by the construction of the code. Let the  $r$  distinct codewords of  $\mathcal{C}^S$  indexing the  $rNl$  columns of  $\mathcal{H}$  be  $\mathcal{A} = \{a_1, a_2, \dots, a_r\}$ . Then the  $rNl \times rNl$  parity-check matrix corresponding to these codewords is given by

$$\mathcal{H}_{\mathcal{A}} = [\mathcal{H}_{a_1} \quad \mathcal{H}_{a_2} \quad \dots \quad \mathcal{H}_{a_r}]$$

$$= \begin{bmatrix} \text{Diag}(I, \dots, I) & \cdots & \text{Diag}(I, \dots, I) \\ \sigma_1 \text{Diag}(H_{a_{1,1}}, \dots, H_{a_{1,N}}) & \cdots & \sigma_r \text{Diag}(H_{a_{r,1}}, \dots, H_{a_{r,N}}) \\ \vdots & \vdots & \vdots \\ \sigma_1^{r-1} \text{Diag}(H_{a_{1,1}}^{r-1}, \dots, H_{a_{1,N}}^{r-1}) & \cdots & \sigma_r^{r-1} \text{Diag}(H_{a_{r,1}}^{r-1}, \dots, H_{a_{r,N}}^{r-1}) \end{bmatrix}.$$

Because of the block diagonal structure of the  $Nl \times Nl$  sub-matrices in the above equation, we need to only show that the following matrix is full-rank, for all  $i \in [N]$ .

$$U_{A,i} = \begin{bmatrix} I & \cdots & I \\ \sigma_1 H_{a_{1,i}} & \cdots & \sigma_r H_{a_{r,i}} \\ \vdots & \vdots & \vdots \\ \sigma_1^{r-1} H_{a_{1,i}}^{r-1} & \cdots & \sigma_r^{r-1} H_{a_{r,i}}^{r-1} \end{bmatrix}$$

$$= \begin{bmatrix} \text{Diag}(1, \dots, 1) & \cdots & \text{Diag}(1, \dots, 1) \\ \sigma_1 \text{Diag}(\lambda_{a_{1,i},f(a_{1,i},0)}, \dots, \lambda_{a_{1,i},f(a_{1,i},l-1)}) & \cdots & \sigma_r \text{Diag}(\lambda_{a_{r,i},f(a_{r,i},0)}, \dots, \lambda_{a_{r,i},f(a_{r,i},l-1)}) \\ \vdots & \vdots & \vdots \\ \sigma_1^{r-1} \text{Diag}(\lambda_{a_{1,i},f(a_{1,i},0)}^{r-1}, \dots, \lambda_{a_{1,i},f(a_{1,i},l-1)}^{r-1}) & \cdots & \sigma_r^{r-1} \text{Diag}(\lambda_{a_{r,i},f(a_{r,i},0)}^{r-1}, \dots, \lambda_{a_{r,i},f(a_{r,i},l-1)}^{r-1}) \end{bmatrix}$$

Re-arranging the rows and columns, we get

$$= \begin{bmatrix} \begin{bmatrix} 1 & \cdots & 1 \\ \sigma_1 \lambda_{a_{1,i},f(a_{1,i},0)} & \cdots & \sigma_r \lambda_{a_{r,i},f(a_{r,i},0)} \\ \vdots & \vdots & \vdots \\ \sigma_1^{r-1} \lambda_{a_{1,i},f(a_{1,i},0)}^{r-1} & \cdots & \sigma_r^{r-1} \lambda_{a_{r,i},f(a_{r,i},0)}^{r-1} \end{bmatrix} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \begin{bmatrix} 1 & \cdots & 1 \\ \sigma_1 \lambda_{a_{1,i},f(a_{1,i},l-1)} & \cdots & \sigma_r \lambda_{a_{r,i},f(a_{r,i},l-1)} \\ \vdots & \vdots & \vdots \\ \sigma_1^{r-1} \lambda_{a_{1,i},f(a_{1,i},l-1)}^{r-1} & \cdots & \sigma_r^{r-1} \lambda_{a_{r,i},f(a_{r,i},l-1)}^{r-1} \end{bmatrix} \end{bmatrix}.$$

$U_{A,i}$  is a block-diagonal matrix where each diagonal block is a Vandermonde matrix. Hence it is a full-rank matrix,  $\forall i \in [N]$  completing the proof. ■

**Remark 1.** It is clear that using  $(n, k, d)$  MSR codes, simultaneous repair of multiple erasures can be performed. However, it is not possible for the case of  $\epsilon$ -MSR codes. This is because for repair of  $\epsilon$ -MSR codes when all the remaining nodes are not contacted, the code has to satisfy  $(P, P')$  repair property and there is a set of  $P'$  compulsory nodes which have to be contacted. If the second erasure is from one of the compulsory nodes, then we cannot recover from the two erasures.

#### IV. REPAIR OF $\epsilon$ -MSCR CODES FOR TWO ERASURES

In this section, we will describe the repair of  $\epsilon$ -MSCR codes for two erasures. We would like to note that as in the case of  $\epsilon$ -MSR codes, an  $\epsilon$ -MSCR code is said to have  $(P, P')$  repair property if for repairing  $h = 2$  erasures,  $P$  nodes are contacted,  $P'$  of which are compulsory nodes and the remaining  $P - P'$  nodes are arbitrary.

Suppose that the nodes  $c^1$  and  $c^2$  indexed by codewords  $a_1, a_2 \in \mathcal{C}^S$  have failed. (9) represents the parity check column  $\mathcal{H}_i$  of  $\mathcal{C}$  corresponding to any codeword  $a_i \in \mathcal{C}^S$ . We give the repair procedure for repairing  $c_1^1$  and  $c_1^2$ . The same procedure can be applied for repairing all  $c_j^1, c_j^2, j \in [2, N]$ , since all parity check columns are block diagonal matrices. Consider the columns of  $\mathcal{H}$  corresponding to  $a_{i,1}, i \in [M]$

$$\begin{bmatrix} I & I & \cdots & I \\ \sigma_1 H_{a_{1,1}} & \sigma_2 H_{a_{2,1}} & \cdots & \sigma_M H_{a_{M,1}} \\ \vdots & \vdots & \cdots & \vdots \\ \sigma_1^{r-1} H_{a_{1,1}}^{r-1} & \sigma_2^{r-1} H_{a_{2,1}}^{r-1} & \cdots & \sigma_M^{r-1} H_{a_{M,1}}^{r-1} \end{bmatrix}.$$

The parity check equation corresponding to  $a_{i,1}, i \in [M]$  is given by

$$\sum_{i=1}^M \sigma_i^t H_{a_{i,1}}^t c_1^i = 0, \quad t \in [0, r-1]. \quad (10)$$

**Remark 2.** We would like to note here that the repair procedure is different for the case when  $a_{1,1} \neq a_{2,1}$  and for the case when  $a_{1,1} = a_{2,1}$ . This is because whenever  $a_{1,1} \neq a_{2,1}$ , then based on the construction of MSCR code, the function  $g(a_{1,1}, a_{2,1})$  denoted as  $g_{12}$  is well defined if  $a_{1,1} < a_{2,1}$ . Otherwise we use the function  $g(a_{2,1}, a_{1,1})$  denoted as  $g_{21}$  and hence the repair is performed based on partitioning the indices according to this function. However, when  $a_{1,1} = a_{2,1}$ , then the function  $g(a_{1,1}, a_{2,1})$  is not defined and for  $a_{3,1} \neq (a_{1,1} = a_{2,1})$ , we perform the repair based on partitioning the indices with respect to  $g(a_{1,1}, a_{3,1})$  denoted as  $g_{13}$  if  $a_{1,1} < a_{3,1}$ . Otherwise we use the function  $g(a_{3,1}, a_{1,1})$  denoted as  $g_{31}$ .

A. Case 1:  $a_{1,1} \neq a_{2,1}$

We construct three disjoint sets  $Q, V, \Gamma$  based on the codewords in  $C^S$  as follows:

$$\begin{aligned} Q &= \{i : a_{i,1} = a_{1,1}, i \in [3, M]\}, \\ V &= \{i : a_{i,1} = a_{2,1}, i \in [3, M]\}, \\ \Gamma &= \{i : a_{i,1} \neq a_{1,1}, a_{i,1} \neq a_{2,1}, i \in [3, M]\}. \end{aligned}$$

$$\sigma_1^t \cdot H_{a_{1,1}}^t \cdot c_1^1 + \sigma_2^t \cdot H_{a_{2,1}}^t \cdot c_1^2 + \sum_{q_i \in Q} \sigma_{q_i}^t \cdot H_{a_{q_i,1}}^t \cdot c_1^{q_i} + \sum_{v_i \in V} \sigma_{v_i}^t \cdot H_{a_{v_i,1}}^t \cdot c_1^{v_i} + \sum_{\gamma_i \in \Gamma} \sigma_{\gamma_i}^t \cdot H_{a_{\gamma_i,1}}^t \cdot c_1^{\gamma_i} = 0, t \in [0, r-1].$$

Considering the parity check equation corresponding to  $b(g_{12}, k)$ ,  $k \in [0, 2]$  and substituting  $\forall q_i \in Q, a_{q_i,1} = a_{1,1}$  and  $\forall v_i \in V, a_{v_i,1} = a_{2,1}$ , we have

$$\begin{aligned} &\sigma_1^t \lambda_{a_{1,1}, f(a_{1,1}, b(g_{12}, k))}^t c_{1, b(g_{12}, k)}^1 + \sigma_2^t \lambda_{a_{2,1}, f(a_{2,1}, b(g_{12}, k))}^t c_{1, b(g_{12}, k)}^2 + \sum_{q_i \in Q} \sigma_{q_i}^t \lambda_{a_{1,1}, f(a_{1,1}, b(g_{12}, k))}^t c_{1, b(g_{12}, k)}^{q_i} \\ &+ \sum_{v_i \in V} \sigma_{v_i}^t \lambda_{a_{2,1}, f(a_{2,1}, b(g_{12}, k))}^t c_{1, b(g_{12}, k)}^{v_i} + \sum_{\gamma_i \in \Gamma} \sigma_{\gamma_i}^t \lambda_{a_{\gamma_i,1}, f(a_{\gamma_i,1}, b(g_{12}, k))}^t c_{1, b(g_{12}, k)}^{\gamma_i} = 0, \quad t \in [0, r-1], k \in [0, 2]. \end{aligned} \quad (11)$$

Applying Claim 1 with  $i = a_{i,1}, i_1 = a_{1,1}, i_2 = a_{2,1}$ , we can define the following:

$$\lambda_{a_{i,1}} := \lambda_{a_{i,1}, f(a_{i,1}, b(g_{12}, 0))} = \lambda_{a_{i,1}, f(a_{i,1}, b(g_{12}, 1))} = \lambda_{a_{i,1}, f(a_{i,1}, b(g_{12}, 2))}, a_{i,1} \notin \{a_{1,1}, a_{2,1}\} \quad (12)$$

$$\lambda'_{a_{1,1}, 0} := \lambda_{a_{1,1}, f(a_{1,1}, b(g_{12}, 0))} = \lambda_{a_{1,1}, f(a_{1,1}, b(g_{12}, 2))}, \quad \lambda'_{a_{1,1}, 1} := \lambda_{a_{1,1}, f(a_{1,1}, b(g_{12}, 1))}, \quad (13)$$

$$\lambda'_{a_{2,1}, 0} := \lambda_{a_{2,1}, f(a_{2,1}, b(g_{12}, 0))} = \lambda_{a_{2,1}, f(a_{2,1}, b(g_{12}, 1))}, \quad \lambda'_{a_{2,1}, 1} := \lambda_{a_{2,1}, f(a_{2,1}, b(g_{12}, 2))}. \quad (14)$$

$\lambda'_{a_{1,1}, 0}, \lambda'_{a_{1,1}, 1}, \lambda'_{a_{2,1}, 0}, \lambda'_{a_{2,1}, 1}, \lambda_{a_{i,1}}, a_{i,1} \notin \{a_{1,1}, a_{2,1}\}$  are all different.

Using the notation defined in (12)-(14), we can write (11) for  $k \in [0, 1]$  and sum over  $k \in [0, 1]$  resulting in

$$\begin{aligned} &\sigma_1^t \sum_{k=0}^1 (\lambda'_{a_{1,1}, k})^t c_{1, b(g_{12}, k)}^1 + \sigma_2^t (\lambda'_{a_{2,1}, 0})^t \sum_{k=0}^1 c_{1, b(g_{12}, k)}^2 + \sum_{q_i \in Q} \sigma_{q_i}^t \sum_{k=0}^1 (\lambda'_{a_{1,1}, k})^t c_{1, b(g_{12}, k)}^{q_i} \\ &+ \sum_{v_i \in V} \sigma_{v_i}^t (\lambda'_{a_{2,1}, 0})^t \sum_{k=0}^1 c_{1, b(g_{12}, k)}^{v_i} + \sum_{\gamma_i \in \Gamma} \sigma_{\gamma_i}^t (\lambda_{a_{\gamma_i,1}})^t \sum_{k=0}^1 c_{1, b(g_{12}, k)}^{\gamma_i} = 0, \quad t \in [0, r-1]. \end{aligned} \quad (15)$$

Let

$$\mu_{2,1,1}^{(b)} = \sum_{k=0}^1 c_{1, b(g_{12}, k)}^2, \quad \mu_{v_i, 1, 1}^{(b)} = \sum_{k=0}^1 c_{1, b(g_{12}, k)}^{v_i}, \quad \mu_{\gamma_i, 1, 1}^{(b)} = \sum_{k=0}^1 c_{1, b(g_{12}, k)}^{\gamma_i}.$$

The eq(15) is of the form

$$L_1 + L_2 + L_3 + L_4 + L_5 = 0, \text{ where} \quad (16)$$

$$\begin{aligned}
L_j &= E_{L_j} F_{L_j}, j \in \{1, 3\}, L_2 = \sum_{q_i \in Q} E_{q_i} F_{q_i}, \\
L_4 &= E_V F_V, L_5 = E_R F_R. \text{ Particularly} \\
E_{L_1} &= [\sigma_1^t (\lambda'_{a_{1,1},k})^t]_{t \in [0, r-1], k \in [0, 1]}, & F_{L_1} &= [c_{1,b(g_{12},k)}^1]_{k \in [0, 1]}, \\
E_{q_i} &= [\sigma_{q_i}^t (\lambda'_{a_{1,1},k})^t]_{t \in [0, r-1], k \in [0, 1]}, & F_{q_i} &= [c_{1,b(g_{12},k)}^{q_i}]_{k \in [0, 1]}, \\
E_{L_3} &= [\sigma_2^t (\lambda'_{a_{2,1},0})^t]_{t \in [0, r-1]}, & F_{L_3} &= \mu_{2,1,1}^{(b)}, \\
E_V &= [\sigma_{v_i}^t (\lambda'_{a_{2,1},0})^t]_{t \in [0, r-1], i \in [1, |V|]}, & F_V &= [\mu_{v_i,1,1}^{(b)}]_{i \in [1, |V|]}, \\
E_R &= [\sigma_{\gamma_i}^t (\lambda_{a_{\gamma_i,1}})^t]_{t \in [0, r-1], i \in [1, |\Gamma|]}, & F_R &= [\mu_{\gamma_i,1,1}^{(b)}]_{i \in [1, |\Gamma|]}.
\end{aligned}$$

We now construct a matrix  $P_1$  and on left-multiplying (16) with it, we get

$$\begin{aligned}
P_1 L_1 + P_1 L_2 + P_1 L_3 + P_1 L_4 + P_1 L_5 &= 0. \\
p_0(x) &= \prod_{k=0}^1 (x - \sigma_1 \lambda'_{a_{1,1},k}) (x - \sigma_2 \lambda'_{a_{2,1},0}) \quad \text{and} \\
p_i(x) &= x^i p_0(x) \quad \text{for } i = 0, 1, \dots, r-4.
\end{aligned}$$

For all  $i \in [0, r-4]$ , the degree of  $p_i(x) < r$ , hence

$$p_i(x) = \sum_{j=0}^{r-1} p_{ij} x^j, \quad i = 0, 1, \dots, r-4.$$

The  $(r-3) \times r$  matrix  $P_1$  is defined as

$$P_1 = \begin{bmatrix} p_{0,0} & p_{0,1} & \cdots & p_{0,r-1} \\ p_{1,0} & p_{1,1} & \cdots & p_{1,r-1} \\ \vdots & \vdots & \vdots & \vdots \\ p_{r-4,0} & p_{r-4,1} & \cdots & p_{r-4,r-1} \end{bmatrix}.$$

$$\begin{aligned}
P_1(L_1 + L_3) &= P_1 \begin{bmatrix} E_{L_1} & E_{L_3} \end{bmatrix} \begin{bmatrix} F_{L_1} \\ F_{L_3} \end{bmatrix} \\
&= P_1 \begin{bmatrix} 1 & 1 & 1 \\ \sigma_1 \lambda'_{a_{1,1},0} & \sigma_1 \lambda'_{a_{1,1},1} & \sigma_2 \lambda'_{a_{2,1},0} \\ \vdots & \vdots & \vdots \\ \sigma_1^{r-1} (\lambda'_{a_{1,1},0})^{r-1} & \sigma_1^{r-1} (\lambda'_{a_{1,1},1})^{r-1} & \sigma_2^{r-1} (\lambda'_{a_{2,1},0})^{r-1} \end{bmatrix} \begin{bmatrix} F_{L_1} \\ F_{L_3} \end{bmatrix} \\
&= \begin{bmatrix} p_0(\sigma_1 \lambda'_{a_{1,1},0}) & p_0(\sigma_1 \lambda'_{a_{1,1},1}) & p_0(\sigma_2 \lambda'_{a_{2,1},0}) \\ \vdots & \cdots & \vdots \\ p_{r-4}(\sigma_1 \lambda'_{a_{1,1},0}) & p_{r-4}(\sigma_1 \lambda'_{a_{1,1},1}) & p_{r-4}(\sigma_2 \lambda'_{a_{2,1},0}) \end{bmatrix} \begin{bmatrix} F_{L_1} \\ F_{L_3} \end{bmatrix} = 0. \\
P_1 L_2 &= P_1 \sum_{q_i \in Q} E_{q_i} F_{q_i} \\
&= \sum_{q_i \in Q} P_1 E_{q_i} F_{q_i} \\
&= \sum_{q_i \in Q} \begin{bmatrix} p_0(\sigma_{q_i} \lambda'_{a_{1,1},0}) & p_0(\sigma_{q_i} \lambda'_{a_{1,1},1}) \\ \vdots & \vdots \\ p_{r-4}(\sigma_{q_i} \lambda'_{a_{1,1},0}) & p_{r-4}(\sigma_{q_i} \lambda'_{a_{1,1},1}) \end{bmatrix} \begin{bmatrix} c_{1,b(g_{12},0)}^{q_i} \\ c_{1,b(g_{12},1)}^{q_i} \end{bmatrix}.
\end{aligned}$$

We can compute  $P_1 L_2$  at the first replacement node by downloading the symbols  $\{c_{1,b(g_{12},0)}^{q_i}, c_{1,b(g_{12},1)}^{q_i}\}$  of  $F_{q_i}, \forall q_i \in Q$ .

$$P_1 L_4 = P_1 E_V F_V$$

$$= \begin{bmatrix} p_0(\sigma_{v_1} \lambda'_{a_{2,1},0}) & \cdots & p_0(\sigma_{v_{|V|}} \lambda'_{a_{2,1},0}) \\ \vdots & \vdots & \vdots \\ p_{r-4}(\sigma_{v_1} \lambda'_{a_{2,1},0}) & \cdots & p_{r-4}(\sigma_{v_{|V|}} \lambda'_{a_{2,1},0}) \end{bmatrix} \begin{bmatrix} \mu_{v_1,1,1}^{(b)} \\ \vdots \\ \mu_{v_{|V|},1,1}^{(b)} \end{bmatrix}.$$

We can compute  $P_1 L_4$  at the first replacement node by downloading the symbols  $\{\mu_{v_i,1,1}^{(b)}\}$  of  $F_V, \forall v_i \in V$ .

$$P_1 L_5 = P_1 \cdot E_R \cdot F_R$$

$$= \begin{bmatrix} p_0(\sigma_{\gamma_1} \lambda_{a_{\gamma_1,1}}) & \cdots & p_0(\sigma_{\gamma_{|\Gamma|}} \lambda_{a_{\gamma_{|\Gamma|},1}}) \\ \vdots & \vdots & \vdots \\ p_{r-4}(\sigma_{\gamma_1} \lambda_{a_{\gamma_1,1}}) & \cdots & p_{r-4}(\sigma_{\gamma_{|\Gamma|}} \lambda_{a_{\gamma_{|\Gamma|},1}}) \end{bmatrix} \begin{bmatrix} \mu_{\gamma_1,1,1}^{(b)} \\ \vdots \\ \mu_{\gamma_{|\Gamma|},1,1}^{(b)} \end{bmatrix}.$$

$p_0(\sigma_{\gamma_1} \lambda_{a_{\gamma_1,1}}) \cdots p_0(\sigma_{\gamma_{|\Gamma|}} \lambda_{a_{\gamma_{|\Gamma|},1}})$  are all non-zero.  $P_1 L_5$  is a full-rank matrix. All  $(r-3) \times (r-3)$  sub-matrices of  $P_1 L_5$  also are full-rank which follows from its Vandermonde like structure. Hence, we have

$$P_1 L_5 = -P_1 L_2 - P_1 L_4. \quad (17)$$

By downloading any subset of size  $k' = (|\Gamma| - (r-3))$  from  $F_R$  ( $|\Gamma| - (r-3) > 0$  since  $q > r$  and  $|\Gamma| \geq q-2$ ), the remaining can be recovered from (17). This is because after substituting  $k'$  values in (17) and rewriting, it would result in  $(r-3)$  equations in  $(r-3)$  variables, which can be solved. From  $(L_1 + L_3) = -L_2 - L_4 - L_5$ , we have

$$\begin{bmatrix} E_{L_1} & E_{L_3} \end{bmatrix} \begin{bmatrix} F_{L_1} \\ F_{L_3} \end{bmatrix} = -L_2 - L_4 - L_5. \quad (18)$$

By inverting a square submatrix of  $\begin{bmatrix} E_{L_1} & E_{L_3} \end{bmatrix}$ , we can recover  $F_{L_1}$  and  $F_{L_3}$ .

We can write an equation similar to (15) for  $k \in \{0, 2\}$  and by performing similar calculations as above, we can recover  $\{c_{1,b(g_{12},0)}^2\} \cup \{c_{1,b(g_{12},2)}^2\} \cup \{\sum_{k=0,2} c_{1,b(g_{12},k)}^1\}$  at the second replacement node.

Please refer to the table for the summary of the downloads and recovery in the two rounds. After both the rounds, first replacement node recovers  $\{c_{1,b(g_{12},k)}^1 : b_{g_{12}} = 0, k \in \{0, 1, 2\}\} = \{c_{1,b}^1 : b \in \{0, 1, \dots, l-1\}\}$  and second replacement node recovers  $\{c_{1,b(g_{12},k)}^2 : b_{g_{12}} = 0, k \in \{0, 1, 2\}\} = \{c_{1,b}^2 : b \in \{0, 1, \dots, l-1\}\}$ . Both  $c_1^1$  and  $c_1^2$  are recovered.

- The repair bandwidth for the case  $a_{1,1} \neq a_{2,1}$  is given by

$$\begin{aligned} RB_{a_{1,1} \neq a_{2,1}} &= k' \frac{l}{3} + |Q| \frac{2l}{3} + |V| \frac{l}{3} + \frac{l}{3} \\ &\quad + k' \frac{l}{3} + |Q| \frac{l}{3} + |V| \frac{2l}{3} + \frac{l}{3} \\ &= k' \left(\frac{2l}{3}\right) + |Q|l + |V|l + \frac{2l}{3}. \end{aligned} \quad (19)$$

where  $k' = |\Gamma| - (r-3)$  and  $|\Gamma| = M - \frac{2M}{q}$ .

- Note that we need to contact all nodes in  $\bar{Q}, V$  compulsorily.

**B. Case 2:**  $a_{1,1} = a_{2,1}$

Consider a node indexed by codeword  $a_3$  in  $\mathcal{C}^S$  such that  $\forall j \in [N], a_{1,j} = a_{2,j}, a_{3,j} \neq (a_{1,j} = a_{2,j})$ . Such a codeword exists in  $\mathcal{C}^S$  if there is a codeword in  $\mathcal{C}^S$  of Hamming weight  $N$ . (The existence of such codewords is guaranteed as we use the same scalar linear code used in [11]). Based on  $a_3$ , we build three sets  $W, Y, Z$  where

$$\begin{aligned} W &= \{i : a_{i,1} = a_{1,1} = a_{2,1}, i \in [4, M]\}, \\ Y &= \{i : a_{i,1} = a_{3,1}, i \in [4, M]\}, \\ Z &= \{i : a_{i,1} \neq (a_{1,1} = a_{2,1}) \ \&\& \ a_{i,1} \neq a_{3,1}, i \in [4, M]\}. \end{aligned}$$

$$\begin{aligned} &\sigma_1^t \cdot H_{a_{1,1}}^t \cdot c_1^1 + \sigma_2^t \cdot H_{a_{2,1}}^t \cdot c_1^2 + \sum_{w_i \in W} \sigma_{w_i}^t \cdot H_{a_{w_i,1}}^t \cdot c_1^{w_i} + \sigma_3^t \cdot H_{a_{3,1}}^t \cdot c_1^3 + \sum_{y_i \in Y} \sigma_{y_i}^t \cdot H_{a_{y_i,1}}^t \cdot c_1^{y_i} \\ &+ \sum_{z_i \in Z} \sigma_{z_i}^t \cdot H_{a_{z_i,1}}^t \cdot c_1^{z_i} = 0, \quad t \in [0, r-1]. \end{aligned} \quad (20)$$



Considering the parity check equation corresponding to  $b(g_{13}, k)$ ,  $k \in [0, 2]$  and substituting  $a_{2,1} = a_{1,1}, \forall w_i \in W, a_{w_i,1} = a_{1,1}$  and  $\forall y_i \in Y, a_{y_i,1} = a_{3,1}$ , we have

$$\begin{aligned} & \sigma_1^t \lambda_{a_{1,1}, f(a_{1,1}, b(g_{13}, k))}^t c_{1, b(g_{13}, k)}^1 + \sigma_2^t \lambda_{a_{1,1}, f(a_{1,1}, b(g_{13}, k))}^t c_{1, b(g_{13}, k)}^2 + \sum_{w_i \in W} \sigma_{w_i}^t \lambda_{a_{1,1}, f(a_{1,1}, b(g_{13}, k))}^t c_{1, b(g_{13}, k)}^{w_i} \\ & + \sigma_3^t \lambda_{a_{3,1}, f(a_{3,1}, b(g_{13}, k))}^t c_{1, b(g_{13}, k)}^3 + \sum_{y_i \in Y} \sigma_{y_i}^t \lambda_{a_{3,1}, f(a_{3,1}, b(g_{13}, k))}^t c_{1, b(g_{13}, k)}^{y_i} \\ & + \sum_{z_i \in Z} \sigma_{z_i}^t \lambda_{a_{z_i,1}, f(a_{z_i,1}, b(g_{13}, k))}^t c_{1, b(g_{13}, k)}^{z_i} = 0, \quad t \in [0, r-1], k \in [0, 2]. \end{aligned} \quad (21)$$

Applying Claim 1 with  $i = a_{i,1}, i_1 = a_{1,1}, i_2 = a_{3,1}$  and using  $g_{13}$  instead of  $g_{12}$ , we can define the following:

$$\lambda_{a_{i,1}} := \lambda_{a_{i,1}, f(a_{i,1}, b(g_{13}, 0))} = \lambda_{a_{i,1}, f(a_{i,1}, b(g_{13}, 1))} = \lambda_{a_{i,1}, f(a_{i,1}, b(g_{13}, 2))}, a_{i,1} \notin \{a_{1,1}, a_{3,1}\} \quad (22)$$

$$\lambda'_{a_{1,1},0} := \lambda_{a_{1,1}, f(a_{1,1}, b(g_{13}, 0))} = \lambda_{a_{1,1}, f(a_{1,1}, b(g_{13}, 2))}, \quad \lambda'_{a_{1,1},1} := \lambda_{a_{1,1}, f(a_{1,1}, b(g_{13}, 1))}, \quad (23)$$

$$\lambda'_{a_{3,1},0} := \lambda_{a_{3,1}, f(a_{3,1}, b(g_{13}, 0))} = \lambda_{a_{3,1}, f(a_{3,1}, b(g_{13}, 1))}, \quad \lambda'_{a_{3,1},1} := \lambda_{a_{3,1}, f(a_{3,1}, b(g_{13}, 2))}. \quad (24)$$

$\lambda'_{a_{1,1},0}, \lambda'_{a_{1,1},1}, \lambda'_{a_{3,1},0}, \lambda'_{a_{3,1},1}, \lambda_{a_{i,1}}, a_{i,1} \notin \{a_{1,1}, a_{3,1}\}$  are all different.

Using the notation defined in (22)-(24), we can write (21) for  $k \in [0, 1]$  and sum over  $k \in [0, 1]$  resulting in

$$\begin{aligned} & \sigma_1^t \sum_{k=0}^1 (\lambda'_{a_{1,1},k})^t c_{1, b(g_{13}, k)}^1 + \sigma_2^t \sum_{k=0}^1 (\lambda'_{a_{1,1},k})^t c_{1, b(g_{13}, k)}^2 + \sigma_3^t (\lambda'_{a_{3,1},0})^t \sum_{k=0}^1 c_{1, b(g_{13}, k)}^3 \\ & + \sum_{w_i \in W} \sigma_{w_i}^t \sum_{k=0}^1 (\lambda'_{a_{1,1},k})^t c_{1, b(g_{13}, k)}^{w_i} + \sum_{y_i \in Y} \sigma_{y_i}^t (\lambda'_{a_{3,1},0})^t \sum_{k=0}^1 c_{1, b(g_{13}, k)}^{y_i} \\ & + \sum_{z_i \in Z} \sigma_{z_i}^t (\lambda_{a_{z_i,1}})^t \sum_{k=0}^1 c_{1, b(g_{13}, k)}^{z_i} = 0, t \in [0, r-1]. \end{aligned} \quad (25)$$

Let

$$\mu_{3,1,1}^{(b)} = \sum_{k=0}^1 c_{1, b(g_{13}, k)}^3, \quad \mu_{y_i,1,1}^{(b)} = \sum_{k=0}^1 c_{1, b(g_{13}, k)}^{y_i}, \quad \mu_{z_i,1,1}^{(b)} = \sum_{k=0}^1 c_{1, b(g_{13}, k)}^{z_i}.$$

(25) can be rewritten as

$$L_1 + L_2 + L_3 + L_4 + L_5 + L_6 = 0, \text{ where} \quad (26)$$

$$\begin{aligned} L_j &= E_{L_j} F_{L_j}, j \in \{1, 3, 4\}, L_2 = \sum_{w_i \in W} E_{w_i} F_{w_i}, \\ L_5 &= E_Y F_Y, L_6 = E_Z F_Z. \text{ Particularly} \\ E_{L_1} &= [\sigma_1^t (\lambda'_{a_{1,1},k})^t]_{t \in [0, r-1], k \in [0, 1]}, & F_{L_1} &= [c_{1, b(g_{13}, k)}^1]_{k \in [0, 1]}, \\ E_{w_i} &= [\sigma_{w_i}^t (\lambda'_{a_{1,1},k})^t]_{t \in [0, r-1], k \in [0, 1]}, & F_{w_i} &= [c_{1, b(g_{13}, k)}^{w_i}]_{k \in [0, 1]}, \\ E_{L_3} &= [\sigma_2^t (\lambda'_{a_{1,1},k})^t]_{t \in [0, r-1], k \in [0, 1]}, & F_{L_3} &= [c_{1, b(g_{13}, k)}^2]_{k \in [0, 1]}, \\ E_{L_4} &= [\sigma_3^t (\lambda'_{a_{3,1},0})^t]_{t \in [0, r-1]}, & F_{L_4} &= \mu_{3,1,1}^{(b)}, \\ E_Y &= [\sigma_{y_i}^t (\lambda'_{a_{3,1},0})^t]_{t \in [0, r-1], i \in [1, |Y|]}, & F_Y &= [\mu_{y_i,1,1}^{(b)}]_{i \in [1, |Y|]}, \\ E_Z &= [\sigma_{z_i}^t (\lambda_{a_{z_i,1}})^t]_{t \in [0, r-1], i \in [1, |Z|]}, & F_Z &= [\mu_{z_i,1,1}^{(b)}]_{i \in [1, |Z|]}. \end{aligned}$$

We now construct a matrix  $P_2$  and on left-multiplying (26) with it, we get

$$\begin{aligned} P_2 L_1 + P_2 L_2 + P_2 L_3 + P_2 L_4 + P_2 L_5 + P_2 L_6 &= 0. \\ p_0(x) &= \prod_{k=0}^1 (x - \sigma_1 \lambda'_{a_{1,1},k}) (x - \sigma_2 \lambda'_{a_{1,1},k}) \quad \text{and} \\ p_i(x) &= x^i p_0(x) \quad \text{for } i = 0, 1, \dots, r-5. \end{aligned}$$

For all  $i \in [0, r-5]$ , the degree of  $p_i(x) < r$ , hence

$$p_i(x) = \sum_{j=0}^{r-1} p_{ij}x^j, \quad i = 0, 1, \dots, r-5.$$

The  $(r-4) \times r$  matrix  $P_2$  is defined as

$$P_2 = \begin{bmatrix} p_{0,0} & p_{0,1} & \cdots & p_{0,r-1} \\ p_{1,0} & p_{1,1} & \cdots & p_{1,r-1} \\ \vdots & \vdots & \vdots & \vdots \\ p_{r-5,0} & p_{r-5,1} & \cdots & p_{r-5,r-1} \end{bmatrix}.$$

For repair, we give a brief description of the steps to be performed at the first replacement node (the procedure is similar to that of the case of  $a_{1,1} \neq a_{2,1}$ ),

- $P_2(L_1 + L_3) = 0$  as  $P_2[E_{L_1} \ E_{L_3}] = 0$ .
- We can compute  $P_2L_2, P_2L_5$  at the first replacement node by downloading the symbols  $\{c_{1,b(g_{13},0)}^{w_i}, c_{1,b(g_{13},1)}^{w_i}\}$  of  $F_{w_i}, \forall w_i \in W, \{\mu_{y_i,1,1}^{(b)}\}$  of  $F_Y, \forall y_i \in Y$  respectively. Then, we have  $P_2(L_4 + L_6) = -P_2L_2 - P_2L_5$ .
- We can compute the remaining symbols of  $\{F_{L_4} \cup F_Z\}$  at the first replacement node by downloading the symbols  $\{c_{1,b(g_{13},0)}^{w_i}, c_{1,b(g_{13},1)}^{w_i}\}$  of  $F_{w_i}, \forall w_i \in W, \{\mu_{y_i,1,1}^{(b)}\}$  symbols of  $F_Y, \forall y_i \in Y$ , any subset of symbols of size  $k'' = (|Z| - (r-5))$  from  $\{F_{L_4} \cup F_Z\}$  ( $|Z| - (r-5) > 0$  since  $q > r$  and  $|Z| \geq q-2$ ), from  $P_2(L_4 + L_6) = -P_2L_2 - P_2L_5$  by inverting a square submatrix of  $P_2[E_{L_4} \ E_Z]$ . Next,  $F_{L_1}$  and  $F_{L_3}$  can be recovered from

$$L_1 + L_3 = -L_2 - L_5 - (L_4 + L_6)$$

by inverting a square submatrix of  $[E_{L_1} \ E_{L_3}]$ .

Similarly using the notation defined in (22)-(24), we can write (21) for  $k = 2$  as follows:

$$L_1 + L_2 + L_3 + L_4 + L_5 + L_6 = 0, \text{ where} \quad (27)$$

$$\begin{aligned} L_j &= E_{L_j}F_{L_j}, j \in \{1, 3, 4\}, L_2 = E_W F_W, \\ L_5 &= E_Y F_Y, L_6 = E_Z F_Z. \text{ Particularly} \\ E_{L_1} &= [\sigma_1^t (\lambda'_{a_{1,1},0})^t]_{t \in [0, r-1]}, & F_{L_1} &= c_{1,b(g_{13},2)}^1, \\ E_W &= [\sigma_{w_i}^t (\lambda'_{a_{1,1},0})^t]_{t \in [0, r-1], i \in [1, |W|]}, & F_W &= [c_{1,b(g_{13},2)}^{w_i}]_{i \in [1, |W|]}, \\ E_{L_3} &= [\sigma_2^t (\lambda'_{a_{1,1},0})^t]_{t \in [0, r-1]}, & F_{L_3} &= c_{1,b(g_{13},2)}^2, \\ E_{L_4} &= [\sigma_3^t (\lambda'_{a_{3,1},1})^t]_{t \in [0, r-1]}, & F_{L_4} &= c_{1,b(g_{13},2)}^3, \\ E_Y &= [\sigma_{y_i}^t (\lambda'_{a_{3,1},1})^t]_{t \in [0, r-1], i \in [1, |Y|]}, & F_Y &= [c_{1,b(g_{13},2)}^{y_i}]_{i \in [1, |Y|]}, \\ E_Z &= [\sigma_{z_i}^t (\lambda_{a_{z_i},1})^t]_{t \in [0, r-1], i \in [1, |Z|]}, & F_Z &= [c_{1,b(g_{13},2)}^{z_i}]_{i \in [1, |Z|]}. \end{aligned}$$

We now construct a matrix  $P_3$  and on left-multiplying (27) with it, we get

$$\begin{aligned} P_3L_1 + P_3L_2 + P_3L_3 + P_3L_4 + P_3L_5 + P_3L_6 &= 0. \\ p_0(x) &= (x - \sigma_1 \lambda'_{a_{1,1},0})(x - \sigma_2 \lambda'_{a_{1,1},0}) \quad \text{and} \\ p_i(x) &= x^i p_0(x) \quad \text{for } i = 0, 1, \dots, r-3. \end{aligned}$$

For all  $i \in [0, r-3]$ , the degree of  $p_i(x) < r$ , hence

$$p_i(x) = \sum_{j=0}^{r-1} p_{ij}x^j, \quad i = 0, 1, \dots, r-3.$$

The  $(r-2) \times r$  matrix  $P_3$  is defined as

$$P_3 = \begin{bmatrix} p_{0,0} & p_{0,1} & \cdots & p_{0,r-1} \\ p_{1,0} & p_{1,1} & \cdots & p_{1,r-1} \\ \vdots & \vdots & \vdots & \vdots \\ p_{r-3,0} & p_{r-3,1} & \cdots & p_{r-3,r-1} \end{bmatrix}.$$

For repair, we give a brief description of the steps to be performed at the second replacement node,

- $P_3(L_1 + L_3) = 0$  as  $P_3[E_{L_1} \ E_{L_3}] = 0$ .
- We can compute  $P_3L_2, P_3L_5$  at the second replacement node by downloading the symbols  $\{c_{1,b(g_{13},2)}^{w_i}\}$  of  $F_{w_i}, \forall w_i \in W, \{c_{1,b(g_{13},2)}^{y_i}\}$  of  $F_Y, \forall y_i \in Y$  respectively. Then, we have  $P_3(L_4 + L_6) = -P_3L_2 - P_3L_5$ .
- We can compute the remaining symbols of  $\{F_{L_4} \cup F_Z\}$  at the second replacement node by downloading the symbols  $\{c_{1,b(g_{13},2)}^{w_i}\}$  of  $F_{w_i}, \forall w_i \in W, \{c_{1,b(g_{13},2)}^{y_i}\}$  symbols of  $F_Y, \forall y_i \in Y$ , any subset of symbols of size  $k''' = (|Z| - (r-3))$  from  $\{F_{L_4} \cup F_Z\}$  ( $|Z| - (r-3) > 0$  since  $q > r$  and  $|Z| \geq q-2$ ), from  $P_3(L_4 + L_6) = -P_3L_2 - P_3L_5$  by inverting a square submatrix of  $P_3[E_{L_4} \ E_Z]$ . Next,  $F_{L_1}$  and  $F_{L_3}$  can be recovered from

$$L_1 + L_3 = -L_2 - L_5 - (L_4 + L_6)$$

by inverting a square submatrix of  $[E_{L_1} \ E_{L_3}]$ .

Please refer to the table for the summary of the downloads and recovery in the two rounds. After both the rounds, first replacement node recovers  $\{c_{1,b(g_{13},k)}^1 : b_{g_{13}} = 0, k \in \{0, 1, 2\}\} = \{c_{1,b}^1 : b \in \{0, 1, \dots, l-1\}\}$  and second replacement node recovers  $\{c_{1,b(g_{13},k)}^2 : b_{g_{13}} = 0, k \in \{0, 1, 2\}\} = \{c_{1,b}^2 : b \in \{0, 1, \dots, l-1\}\}$ . Both  $c_1^1$  and  $c_1^2$  are recovered.

1) The repair bandwidth for the case  $a_{1,1} = a_{2,1}$  is given by

$$\begin{aligned} RB_{a_{1,1}=a_{2,1}} &= k'' \frac{l}{3} + |W| \frac{2l}{3} + |Y| \frac{l}{3} + \frac{l}{3} \\ &\quad + k''' \frac{l}{3} + |W| \frac{l}{3} + |Y| \frac{l}{3} + \frac{2l}{3} \\ &= (k'' + k''') \frac{l}{3} + |W|l + |Y| \frac{2l}{3} + l. \end{aligned} \quad (28)$$

where  $k'' = |Z| - (r-5)$ ,  $k''' = |Z| - (r-3)$  and  $|Z| = M - \frac{2M}{q}$ .

2) Note that we need to compulsorily contact all the nodes given by the sets  $W, Y$ .

	1 <sup>st</sup> replacement node	2 <sup>nd</sup> replacement node
R1: $a_{1,1} \neq a_{2,1}$	Download: $\{\mu_{i,1}^{(b)} : b_{g_{12}} = 0\}$ from $k'$ of $ \Gamma $ nodes $\{c_{1,b(g_{12},0)}^{q_i}, c_{1,b(g_{12},1)}^{q_i} : b_{g_{12}} = 0, \forall q_i \in Q,$ $\{\sum_{k=0}^1 c_{1,b(g_{12},k)}^{v_i} : b_{g_{12}} = 0, \forall v_i \in V.$ Recovery: $\{c_{1,b(g_{12},0)}^1 : b_{g_{12}} = 0\}$ $\cup \{c_{1,b(g_{12},1)}^1 : b_{g_{12}} = 0\} \cup \{\mu_{2,1}^{(b)} : b_{g_{12}} = 0\}.$	Download: $\{\mu_{i,2}^{(b)} : b_{g_{12}} = 0\}$ from $k'$ of $ \Gamma $ nodes $\{c_{1,b(g_{12},0)}^{v_i}, c_{1,b(g_{12},2)}^{v_i} : b_{g_{12}} = 0, \forall v_i \in V,$ $\{\sum_{k=0,2} c_{1,b(g_{12},k)}^{q_i} : b_{g_{12}} = 0, \forall q_i \in Q.$ Recovery: $\{c_{1,b(g_{12},0)}^2 : b_{g_{12}} = 0\}$ $\cup \{c_{1,b(g_{12},2)}^2 : b_{g_{12}} = 0\} \cup \{\mu_{1,2}^{(b)} : b_{g_{12}} = 0\}.$
R2: $a_{1,1} \neq a_{2,1}$	Download: $\{\mu_{1,2}^{(b)} : b_{g_{12}} = 0\}.$	Download: $\{\mu_{2,1}^{(b)} : b_{g_{12}} = 0\}.$
R1: $a_{1,1} = a_{2,1}$	Download: $\{\alpha_{i,1}^{(b)} : b_{g_{13}} = 0\}$ from $k''$ of $ Z' $ nodes $\{c_{1,b(g_{13},0)}^{w_i}, c_{1,b(g_{13},1)}^{w_i} : b_{g_{13}} = 0\} \forall w_i \in W,$ $\{\sum_{k=0}^1 c_{1,b(g_{13},k)}^{y_i} : b_{g_{13}} = 0\} \forall y_i \in Y.$ Recovery: $\{c_{1,b(g_{13},0)}^1 : b_{g_{13}} = 0\}$ $\cup \{c_{1,b(g_{13},1)}^1 : b_{g_{13}} = 0\} \cup \{c_{1,b(g_{13},0)}^2 : b_{g_{13}} = 0\}$ $\cup \{c_{1,b(g_{13},1)}^2 : b_{g_{13}} = 0\}.$	Download: $\{c_{1,b(g_{13},2)}^i : b_{g_{13}} = 0\}$ from $k'''$ of $ Z' $ , $\{c_{1,b(g_{13},2)}^{w_i} : b_{g_{13}} = 0\} \forall w_i \in W,$ $\{c_{1,b(g_{13},2)}^{y_i} : b_{g_{13}} = 0\} \forall y_i \in Y.$ Recovery: $\{c_{1,b(g_{13},2)}^1 : b_{g_{13}} = 0\}$ $\cup \{c_{1,b(g_{13},2)}^2 : b_{g_{13}} = 0\}.$
R2: $a_{1,1} = a_{2,1}$	Download: $\{c_{1,b(g_{13},2)}^1 : b_{g_{13}} = 0\}.$	Download: $\{c_{1,b(g_{13},0)}^2, c_{1,b(g_{13},1)}^2 : b_{g_{13}} = 0\}.$

TABLE I

SUMMARY OF DOWNLOADS AND RECOVERY IN THE TWO ROUNDS OF COOPERATIVE REPAIR, WHERE  $\mu_{i,1}^{(b)} = c_{1,b(g_{12},0)}^i + c_{1,b(g_{12},1)}^i$ ,  
 $\mu_{i,2}^{(b)} = c_{1,b(g_{12},0)}^i + c_{1,b(g_{12},2)}^i$ ,  $\alpha_{i,1}^{(b)} = c_{1,b(g_{13},0)}^i + c_{1,b(g_{13},1)}^i$  AND  $Z' = Z \cup \{3\}$ .

## V. REPAIR BANDWIDTH ANALYSIS OF $\epsilon$ -MSCR CODES

In this section, we give the repair bandwidth for the repair method described in Section IV and compare it with that of the optimal. We also characterize the sub-packetization level.

**Lemma V.1.** *The number of nodes  $P$  contacted for the repair of two erasures (described in Section III) is atleast  $(M - r)$ .*

*Proof:* For the case of  $a_{1,1} \neq a_{2,1}$ , number of nodes contacted is

$$\begin{aligned} &\geq |Q| + |V| + |\Gamma| - (r - 3) \\ &\geq M - 2 - r + 3 \\ &\geq M - r. \end{aligned}$$

For  $a_{1,1} = a_{2,1}$ , number of nodes contacted is

$$\begin{aligned} &\geq |W| + |Y| + |Z| - (r - 5) \\ &\geq M - 3 - r + 5 \\ &\geq M - r. \end{aligned}$$

■

**Remark 3.** *It is not straight-forward to characterize the number of compulsory nodes in terms of the hamming weight of the individual codewords as in [11]. Hence, we leave it for future work.*

**Theorem V.2.** *The repair bandwidth for the method described in Section IV is at most  $(1 + \epsilon)$  times away from the optimal repair bandwidth where  $\epsilon \leq (\frac{r}{P+1})(\frac{1}{2} + (2 - \delta)\frac{P}{3}) - 1$ ,  $P$  is the total number of nodes contacted for the repair of the two failed nodes.*

*Proof:* Let  $\mathcal{P}$  denote the set of contacted nodes for repair and let  $|\mathcal{P}| = P$ . For a given node  $i$  and each  $j \in [N]$ , if we assume that  $a_{i,j}$  is helping the repair process via one of the sets  $(Q, V, \Gamma, W, Y, Z)$ , then we have the following upper bound on the total repair bandwidth.

$$\begin{aligned} RB_{tot} &\leq |\{j \in [1, N] : a_{1,j} \neq a_{2,j}\}| \frac{2l}{3} + |\{j \in [1, N] : a_{1,j} = a_{2,j}\}| l + \sum_{i \in \mathcal{P}} (|\{j \in [1, N] : a_{1,j} \neq a_{2,j}, a_{i,j} = a_{1,j}\}| l \\ &\quad + |\{j \in [1, N] : a_{1,j} \neq a_{2,j}, a_{i,j} = a_{2,j}\}| l + |\{j \in [1, N] : a_{1,j} \neq a_{2,j}, a_{i,j} \neq a_{1,j}, a_{i,j} \neq a_{2,j}\}| \frac{2l}{3} \\ &\quad + |\{j \in [1, N] : a_{1,j} = a_{2,j}, a_{i,j} = a_{1,j}\}| l + |\{j \in [1, N] : a_{1,j} = a_{2,j}, a_{i,j} = a_{3,j}\}| \frac{2l}{3} \\ &\quad + |\{j \in [1, N] : a_{1,j} = a_{2,j}, a_{i,j} \neq a_{1,j}, a_{i,j} \neq a_{3,j}\}| \frac{2l}{3}) \end{aligned} \quad (29)$$

where the first two terms correspond to the repair bandwidth in the second round for  $a_{1,1} \neq a_{2,1}$  and  $a_{1,1} = a_{2,1}$  respectively. We note that

$$\begin{aligned} &|\{j \in [1, N] : a_{1,j} \neq a_{2,j}\}| \frac{2l}{3} + |\{j \in [1, N] : a_{1,j} = a_{2,j}\}| l \\ &\quad \leq |\{j \in [1, N] : a_{1,j} \neq a_{2,j}\}| l + |\{j \in [1, N] : a_{1,j} = a_{2,j}\}| l \\ &\quad \leq Nl. \end{aligned} \quad (30)$$

and also

$$\begin{aligned} &|\{j \in [1, N] : a_{1,j} \neq a_{2,j}, a_{i,j} \neq a_{1,j}, a_{i,j} \neq a_{2,j}\}| \\ &\quad = N - |\{j \in [1, N] : a_{1,j} \neq a_{2,j}, a_{i,j} = a_{1,j}\}| - |\{j \in [1, N] : a_{1,j} \neq a_{2,j}, a_{i,j} = a_{2,j}\}| \\ &\quad \quad - |\{j \in [1, N] : a_{1,j} = a_{2,j}, a_{i,j} = a_{1,j}\}| - |\{j \in [1, N] : a_{1,j} = a_{2,j}, a_{i,j} = a_{3,j}\}| \\ &\quad \quad - |\{j \in [1, N] : a_{1,j} = a_{2,j}, a_{i,j} \neq a_{1,j}, a_{i,j} \neq a_{3,j}\}|. \end{aligned} \quad (31)$$

Substituting (30), (31) in (29), we have

$$\begin{aligned} RB_{tot} &\leq Nl + \sum_{i \in \mathcal{P}} \left( \frac{2Nl}{3} + |\{j \in [1, N] : a_{1,j} \neq a_{2,j}, a_{i,j} = a_{1,j}\}| \frac{l}{3} + |\{j \in [1, N] : a_{1,j} \neq a_{2,j}, a_{i,j} = a_{2,j}\}| \frac{l}{3} \right. \\ &\quad \left. + |\{j \in [1, N] : a_{1,j} = a_{2,j}, a_{i,j} = a_{1,j}\}| \frac{l}{3} \right) \end{aligned} \quad (32)$$

$$RB_{tot} \leq Nl + \sum_{i \in \mathcal{P}} \left( \frac{2Nl}{3} + |\{j \in [1, N] : a_{i,j} = a_{1,j}\}| \frac{l}{3} + |\{j \in [1, N] : a_{1,j} \neq a_{2,j}, a_{i,j} = a_{2,j}\}| \frac{l}{3} \right) \quad (33)$$

$$\begin{aligned}
&\leq Nl + \sum_{i \in \mathcal{P}} \left( \frac{2Nl}{3} + 2(N-D)\left(\frac{l}{3}\right) \right) \\
&\leq Nl + \sum_{i \in \mathcal{P}} \left( \frac{4Nl}{3} - \frac{2Dl}{3} \right).
\end{aligned} \tag{34}$$

From (2), the optimal repair bandwidth for co-operative repair is

$$\begin{aligned}
(RB)_{opt} &= \frac{h(h+d-1)l}{h+d-k} \\
&= \frac{h(h+P-1)L}{h+P-\frac{K_S}{Nl}} \\
&= \frac{2(P+1)Nl}{2+P-(M-r)}.
\end{aligned} \tag{35}$$

as  $h = 2, K_S = (M-r)Nl, d = P, L = Nl$  for our code. Since  $P \leq M-2$ , we have

$$\frac{2(P+1)Nl}{P-(M-r)+2} \geq \frac{2(P+1)Nl}{M-2-(M-r)+2}. \tag{36}$$

To derive an upper bound on  $\epsilon$ , we consider the following equations:

$$Nl + P\left(\frac{4Nl}{3} - \frac{2Dl}{3}\right) = (1+\epsilon)\left(\frac{2(P+1)Nl}{P-(M-r)+2}\right) = (1+\epsilon_2)\left(\frac{2(P+1)Nl}{M-2-(M-r)+2}\right). \tag{37}$$

From (36) and (37),  $\epsilon \leq \epsilon_2$ . Also, considering first and third terms from (37), we have

$$\begin{aligned}
Nl + P\left(\frac{4Nl}{3} - \frac{2Dl}{3}\right) &= (1+\epsilon_2)\left(\frac{2(P+1)Nl}{r}\right) \\
\implies \frac{1}{2} + P\left(\frac{2}{3} - \frac{\delta}{3}\right) &= \frac{(1+\epsilon_2)(P+1)}{r} \\
\implies \epsilon_2 &= \frac{r}{P+1}\left(\frac{1}{2} + (2-\delta)\frac{P}{3}\right) - 1.
\end{aligned}$$

Since  $\epsilon \leq \epsilon_2$ , we have

$$\epsilon \leq \left(\frac{r}{P+1}\right)\left(\frac{1}{2} + (2-\delta)\frac{P}{3}\right) - 1. \tag{38}$$

Hence, the repair bandwidth of the repair method that we described in Section IV is atmost  $(1+\epsilon) = \left(\frac{r}{P+1}\right)\left(\frac{1}{2} + (2-\delta)\frac{P}{3}\right)$  times away from the optimal repair bandwidth completing the proof. ■

**Corollary V.3.** For the case of  $r = 5$ , our construction with the given repair procedure results in an  $\epsilon$ -MSCR code whose repair bandwidth is  $(1+\epsilon) = \frac{5}{6} \left( \frac{3+(2-\delta)2P}{P+1} \right) (\sim \frac{5}{3}$ , when  $\delta$  is close to 1 and  $P$  is large) times away from the optimal repair bandwidth.

**Theorem V.4.** Given positive integers  $r, q, u$  and an  $\epsilon > 0$ , there exists an  $(M, K_S = (M-r)Nl, d_{min} = r+1, L = Nl)_{\mathbb{B}}$   $\epsilon$ -MSCR code satisfying the  $(\tau, \tau')$  repair property with sub-packetization scaling logarithmically with  $M$  for constant  $q, u$  and the required field size of the order of  $M$  for constant  $q$ .

*Proof:* An  $\epsilon$ -MSCR code with length  $M = q^K = q^{ug}$  and  $L = Nl = N3^m$ ,  $m = \binom{n}{2}$  is obtained by combining a  $(n = q, k = q-r, l = 3^m)_{\mathbb{B}}$ MSCR code with a scalar linear code  $\mathcal{C}^S = (N, M = q^K, D = \delta N)_q$  with  $K = ug$  and  $\frac{g}{N} \approx \frac{1}{\sqrt{q}-1}$  (the choice of these parameters are obtained from Theorem 3.3 in [11]).

$$M = q^{ug}.$$

By taking log on both sides and using  $g \approx \frac{N}{\sqrt{q}-1}$ , we get

$$\log M = \frac{uN}{\sqrt{q}-1} \log q.$$

But  $L = N3^{\binom{n}{2}} = N3^{\binom{q}{2}}$ , so we get

$$\log M = \frac{uL}{3^{\binom{q}{2}}(\sqrt{q}-1)} \log q.$$

Since  $q, u$  are constant, we have  $L = O_{q,u}(\log M)$ .

To have distinct  $\{\lambda_{i,j}\}_{i \in [n=q], j \in [0,1]}$  for construction of MSCR code, we need a field size of at-least  $2q$ . Next for distinct scalars  $\{\sigma_i\}_{i \in [M]}$  for construction of  $\epsilon$ -MSCR code, we need a field size of at-least  $M = q^K$ . So, overall for the construction of  $\epsilon$ -MSCR code, we need a field size of at-least  $2q^K q + 1$ . For constant  $q$ , the required field size is  $O_q(q^K) = O_q(M)$  which proves the above theorem. ■

#### REFERENCES

- [1] A. G. Dimakis, P. B. Godfrey, Y. Wu, M. J. Wainwright, and K. Ramchandran, "Network coding for distributed storage systems," *IEEE transactions on information theory*, vol. 56, no. 9, pp. 4539–4551, 2010.
- [2] K. V. Rashmi, N. B. Shah, and P. V. Kumar, "Optimal exact-regenerating codes for distributed storage at the MSR and MBR points via a product-matrix construction," *IEEE Transactions on Information Theory*, vol. 57, no. 8, pp. 5227–5239, 2011.
- [3] D. S. Papailiopoulos, A. G. Dimakis, and V. R. Cadambe, "Repair optimal erasure codes through hadamard designs," *IEEE Transactions on Information Theory*, vol. 59, no. 5, pp. 3021–3037, 2013.
- [4] V. R. Cadambe, S. A. Jafar, H. Maleki, K. Ramchandran, C. Suh, *et al.*, "Asymptotic interference alignment for optimal repair of mds codes in distributed storage," *IEEE Trans. Informations on Information Theory*, vol. 59, no. 5, pp. 2974–2987, 2013.
- [5] I. Tamo, Z. Wang, and J. Bruck, "Zigzag codes: MDS array codes with optimal rebuilding," *IEEE Transactions on Information Theory*, vol. 59, no. 3, pp. 1597–1616, 2013.
- [6] M. Ye and A. Barg, "Explicit constructions of high-rate MDS array codes with optimal repair bandwidth," *IEEE Trans. Information Theory*, vol. 63, no. 4, pp. 2001–2014, 2017.
- [7] M. Ye and A. Barg, "Explicit constructions of optimal-access MDS codes with nearly optimal sub-packetization," *IEEE Transactions on Information Theory*, vol. 63, no. 10, pp. 6307–6317, 2017.
- [8] 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," *arXiv preprint arXiv:1607.07335*, 2016.
- [9] S. Goparaju, I. Tamo, and R. Calderbank, "An improved sub-packetization bound for minimum storage regenerating codes," *IEEE Transactions on Information Theory*, vol. 60, no. 5, pp. 2770–2779, 2014.
- [10] A. S. Rawat, I. Tamo, V. Guruswami, and K. Efremenko, "MDS code constructions with small sub-packetization and near-optimal repair bandwidth," *IEEE Transactions on Information Theory*, vol. 64, no. 10, pp. 6506–6525, 2018.
- [11] V. Guruswami, S. V. Lokam, and S. V. M. Jayaraman, " $\epsilon$ -MSR codes: Contacting fewer code blocks for exact repair," *arXiv preprint arXiv:1807.01166*, 2018.
- [12] K. W. Shum and Y. Hu, "Cooperative regenerating codes," *IEEE Transactions on Information Theory*, vol. 59, no. 11, pp. 7229–7258, 2013.
- [13] K. W. Shum and J. Chen, "Cooperative repair of multiple node failures in distributed storage systems," *International Journal of Information and Coding Theory*, vol. 3, no. 4, pp. 299–323, 2016.
- [14] J. Li and B. Li, "Cooperative repair with minimum-storage regenerating codes for distributed storage," in *INFOCOM, 2014 Proceedings IEEE*, pp. 316–324, IEEE, 2014.
- [15] M. Ye and A. Barg, "Cooperative repair: Constructions of optimal MDS codes for all admissible parameters," *IEEE Transactions on Information Theory*, 2018.
- [16] Y. Zhang and Z. Zhang, "Scalar MSCR codes via the product matrix construction," *arXiv preprint arXiv:1810.04611*, 2018.
- [17] M. Zorghi and Z. Wang, "Centralized multi-node repair for minimum storage regenerating codes," in *Information Theory (ISIT), 2017 IEEE International Symposium on*, pp. 2213–2217, IEEE, 2017.