Optimal Busy-Node Repair of $(k+4, k, 4)$ MDS Codes with Small Sub-packetization Level

Jiayi Rui, Qin Huang, *Senior Member, IEEE*, Yunghsiang S. Han, *Fellow, IEEE*, Ting-Yi Wu

***Abstract*—Maximum distance separable array codes with small sub-packetization level (MDS-SSL) are of practical importance in distributed storage systems. This letter addresses their repair of a single failed node in the case that there exists a busy node during the repair. Additional check relations among codeword symbols excluding those in the busy node are formed to compensate for the absence of information. Then, it allows us to repair the failed node without symbols in the busy node. A lower bound of the busy-node repair bandwidth is derived for $(k+4, k, 4)$ MDS-SSL codes, consisting of four groups of nodes. It indicates that its achievability requires a cubic constraint among elements in the parity-check matrix. Then, an explicit busy-node repair scheme is proposed to achieve a lower bound in the case of the failed node and the busy node being in different groups.**

***Index Terms*—Distributed storage system, busy-node repair, optimal access, small sub-packetization, bandwidth.**

I. Introduction

In a distributed storage system with (n,k,l) maximum distance separable (MDS) codes, all symbols stored in n nodes can be recovered by connecting to k arbitrary nodes. The data of each node is partitioned into l sections, where l is referred to as sub-packetization level or node size. Small sub-packetization level (e.g., $l = 4$) [1], [2] not only contributes to an easy implementation of the system, but also may distribute the load of providing information for the repair of a failed node among a large number of nodes. Thus, *MDS codes with small sub-packetization level* (MDS-SSL) are required in large scale distributed storage systems. Recently, Rawat proposed $(n, k, l = n-k)$ MDS-SSL with near optimal repair bandwidth for single node repair in [2]. Such MDS-SSL codes have attracted much attention in both academia and industry.

In the single node repair model of [2]–[7], nodes involved in the node repair are all accessible. However, some nodes may be busy during the repair process in practice. Most repairing schemes for a single-node failure need to access the remaining $n-1$ nodes, which is not applicable when some survival nodes are busy. It should be mentioned that, unlike the multi-node repair [8] needs to recover all inaccessible nodes, the busy-node repair only needs to recover the failed node while some busy nodes are inaccessible. The difficulty of busy-node repair is that we need to optimize both performances of repairing a single node with or without busy nodes simultaneously. In particular, the majority of all busy cases are one busy node.

Since the information in the busy node is unavailable, the repair scheme designed in need of $n-1$ surviving nodes, cannot provide enough check relations for the failed node repair. This letter proposes to form additional check relations among codeword symbols excluding those in the busy node to compensate for the absence of information in the busy node. We focus on the family of $(n = k+4, k, l = 4)$ MDS-SSL codes which are of practical importance [2]. Such codes divide the nodes into l groups, according to the structure of the generator matrix of each node. A lower bound of the busy-node repair bandwidth is derived by going through all cases. It indicates that its achieveability requires a *cubic constraint* among elements in the parity-check matrix. Then a busy-node repair scheme under such constraint is proposed to achieve a lower bound in the case of the failed node and the busy node being in different groups.

This work was supported by the National Natural Science Foundation of China under Grant 62071026 and 61941106.

Jiayi Rui and Qin Huang are with the School of Electronic and Information Engineering, Beihang University, Beijing 100191, China (email: jiayirui@buaa.edu.cn; qhuang.smash@gmail.com). *(Corresponding author: Qin Huang.)*

Yunghsiang S. Han is with the Shenzhen Institute for Advanced Study, University of Electronic Science and Technology of China (email: yunghsiangh@gmail.com).

Ting-Yi Wu is with the Theory Lab, Central Research Institute, 2012 Labs, Huawei Technology Co. Ltd (email: wu.ting.yi@huawei.com).

II. Preliminary

Consider (n, k, l) MDS codes for a distributed storage system with k data nodes and $n-k$ redundant nodes. Each node store l symbols and the l symbols stored in a node are viewed as an l length vector. For a positive integer t, $[t]$ denotes the set $\{1, 2, \ldots, t\}$. Let $|U|$ denote the cardinality of the set U. The following illustrates the construction of $(n, k, l = n-k)$ MDS-SSL codes [2].

Let $r = n-k = l$ and $n = sr$. The n nodes are partitioned into r groups with size s in each. Each node is indexed by a tuple (u, v), where $u \in [r]$ and $v \in [s]$. In particular, for $i \in [n]$, the associated tuple (u, v) satisfies $i = (u-1)s+v$. With this notation in place, for $(u, v) \in [r]\times[s]$, the $((u-1)s+v)$-th node is denoted by

$$\mathbf{c}_{(u-1)s+v} = \mathbf{c}_{(u,v)} = (c(1;(u,v)), \ldots, c(r;(u,v))) \in \mathbb{B}^r,$$

where for $x \in [r]$, $c(x;(u,v))$ denotes the x-th symbol of the $((u-1)s+v)$-th node and the finite field $\mathbb{B}$ is an extension field of the finite filed $\mathbb{L}$ of size at least $n+1$. Let $\{\lambda_i\}_{i\in[n]}$ be n distinct nonzero elements of $\mathbb{L}$ and $\{\psi_{i,j}\}_{i\in[n],j\in[r-1]}$ be elements of $\mathbb{B}$. We can present the $r^2 \times nr$ parity-check matrix of the code [2] with parameters $(n, k, l = r = n-k)$ as

$$\mathbf{H} = \begin{bmatrix} \mathbf{I} & \mathbf{I} & \cdots & \mathbf{I} \\ \lambda_1\mathbf{I}+\Psi_{1,1} & \lambda_2\mathbf{I}+\Psi_{2,1} & \cdots & \lambda_n\mathbf{I}+\Psi_{n,1} \\ \vdots & \vdots & \cdots & \vdots \\ \lambda_1^{r-1}\mathbf{I}+\Psi_{1,r-1} & \lambda_2^{r-1}\mathbf{I}+\Psi_{2,r-1} & \cdots & \lambda_n^{r-1}\mathbf{I}+\Psi_{n,r-1} \end{bmatrix}, \quad (1)$$

where $\mathbf{I}$ is the $r \times r$ identity matrix, $\Psi_{(u-1)s+v,p}$ is the $r \times r$ matrix with the element $\psi_{(u-1)s+v,p}$ in the position $(u, \overline{u+p})$ and zeros elsewhere, $u \in [r], v \in [s]$, $p \in [r-1]$ and for a strictly positive integer e, the quantity $\bar{e}$ is defined as follows:

$$\bar{e} = \begin{cases} r & \text{if } e \pmod r = 0, \\ e \pmod r & \text{otherwise.} \end{cases} \tag{2}$$

Let $\mathbf{H} = \begin{bmatrix} \mathrm{H}_0^{\mathrm{T}} & \mathrm{H}_1^{\mathrm{T}} & \cdots & \mathrm{H}_{r-1}^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}}$ and $\mathrm{H}_i = \begin{bmatrix} h_{i,1}^{\mathrm{T}} & h_{i,2}^{\mathrm{T}} & \cdots & h_{i,r}^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}}$, where H_i's are matrices of size $r \times nr$, $h_{i,j}$'s are vectors of length nr, $0 \le i \le r-1$, $1 \le j \le r$ and T denotes the transpose operation.

Then we illustrate the exact repair of a node. Let $(u^*, v^*) \in [r] \times [s]$ be the tuple associated with the failed node. Let

$$U_f = \{c(x; (u^*, v^*)) | x \in [r]\}$$

denote the set of r symbols which need to be recovered. The repair scheme takes advantage of r linear constraints defined by r rows $\{h_{0,u^*}, \ldots, h_{r-1,u^*}\}$ of the parity-check matrix, i.e., $\{h_{i,u^*} \cdot [\mathbf{c}_1 \ldots \mathbf{c}_n]^{\mathrm{T}} | i = 0, 1, 2, 3\}$, as below. For every $p \in [r-1]$,

$$\sum_{(u,v) \in [r] \times [s]} c(u^*; (u, v)) = 0; \tag{3}$$

$$\begin{aligned} &\sum_{(u,v) \in [r] \times [s]} \lambda^p_{(u-1)s+v} \cdot c(u^*; (u, v)) \\ &+ \sum_{v \in [s]} \psi_{(u^*-1)s+v,p} \cdot c(\overline{u^*+p}; (u^*, v)) = 0. \end{aligned} \tag{4}$$

These linear constraints show that the set of symbols involved in Eq. (3) is $A_0 = \{c(u^*; (u, v)) | (u, v) \in [r] \times [s]\}$. Sets of symbols involved in Eq. (4) are $A_p = A_0 \cup \{c(\overline{u^*+p}; (u^*, v)) | v \in [s]\}$, for every $p \in [r-1]$. Eqs. (3) and (4) indicate that there are r unknown symbols (i.e., r symbols to be repaired) and r equations. In order to recover the failed node, it needs to download the set of symbols as

$$U_{nb} = (A_0 \cup \cdots \cup A_{r-1}) \backslash U_f,$$

where the subscript "nb" means the condition that there is no busy node. Note that the total number of symbols to be downloaded is $|U_{nb}| = r(s-1) + s(r-1) = 2n - s - r$ which is referred to as repair bandwidth and denoted by γ_{nb}.

III. Lower Bound of Busy-Node Repair Bandwidth

Under the repair scheme in Section II, all $n-1$ remaining nodes are required in a failed node repair. It cannot repair the failed node once one remaining node is busy. Thus, a busy-node repair scheme is necessary in consideration of keeping the repair scheme when there is no busy node.

In this section, we analyze a lower bound of the busy-node repair bandwidth under the case of $(n, k, l = r = n - k = 4)$ and $s = n/r > 7$. In many practical systems, nodes have no computation ability such that we only consider access bandwidth, i.e., the average of the number of access symbols involved in recovering a single node. Suppose that the node with the tuple (u^*, v^*) fails and the node with the tuple (u', v') is busy, where $(u^*, v^*), (u', v') \in [r] \times [s]$ and $(u', v') \neq (u^*, v^*)$. Since the node with the tuple (u', v') is busy, the symbols (in the busy node) involved in the repair of the failed node are unavailable. Eqs. (3) and (4) turn to be more than r unknown symbols (symbols in the failed node and busy node) and r equations. The node with the tuple (u^*, v^*) cannot be recovered. Thus, we propose to form additional equations (i.e., linear constraints) by taking linear combinations of rows of $\mathbf{H}$ except rows used in Eqs. (3) and (4).

Suppose that additional equations (i.e., linear constraints) are defined by distinct vectors $E_1, \ldots, E_m$ which are linear combinations of rows of $\mathbf{H}$ except rows used in Eqs. (3) and (4), where $1 \le m \le |\mathbb{B}|^{12} - 1$. Linear constraints defined by $\{E_1, \ldots, E_m\}$ together with Eqs. (3) and (4) form the whole busy-node repair. Let U_i denote the set of symbols involved in linear constraints defined by E_i, where $i \in [m]$. For $\forall A = \{i_1, \ldots, i_t\} \subset [m]$, linear constraints defined by $\{E_{i_1}, \ldots, E_{i_t}\}$ lead to download the set of symbols $U_A = (U_{i_1} \cup \cdots \cup U_{i_t}) \backslash (U_f \cup U_b)$, where $1 \le t \le m$, A is a non-empty set and $U_b = \{c(x; (u', v')) | x \in [r]\}$ denotes the set of symbols in the busy node. The additional bandwidth brought by $\{E_{i_1}, \ldots, E_{i_t}\}$ is defined as $\gamma_A = |U_A \backslash U_{nb}|$. Specially, let γ denote additional bandwidth brought by $\{E_1, \ldots, E_m\}$ for simplicity. According to definitions, it is obvious that for $\forall A \subset [m]$

$$\gamma_A \le \gamma \le \sum_{i=1}^{m} \gamma_i, \tag{5}$$

where γ_i is the shorthand notation of $\gamma_{\{i\}}$.

A lower bound of busy-node repair bandwidth is derived by analyzing additional bandwidth brought by additional equations. Before the analysis of lower bound of the busy-node repair bandwidth, we first present the parity-check matrix $\mathbf{H}$ in the form as shown in Fig. 1, which is obtained by employing row and column permutations on the parity-check matrix $\mathbf{H}$. Correspondences between columns and symbols are illustrated in Fig. 1. This form helps for the illustration of the analysis.

A. Failed node and busy node are in different groups

In this subsection, we consider the case that the failed node and busy node are in different groups i.e., $u' \neq u^*$. The set of symbols downloaded in Eqs. (3) and (4) is $U_{nb} \backslash \{c(u^*; (u', v'))\}$. Eqs. (3) and (4) turn to be $r+1$ unknown symbols, $U_f \cup \{c(u^*; (u', v'))\}$, and r equations. Thus, at least another one equation is required. According to Eq. (5), we have $\gamma \ge \gamma_1$. The analysis of γ starts from additional bandwidth γ_1 brought by one vector E_1. Then consider whether E_2 is necessary. If E_2 is unnecessary, Eqs. (3), (4) together with E_1 form a busy-node repair scheme; otherwise, analyze additional bandwidth brought by vectors $\{E_1, E_2\}$. Repeat the above process until we can form a busy-node repair scheme. A lower bound of busy-node repair bandwidth is derived from going through all cases.

The analysis of γ_1 is equal to analyze nonzero elements in E_1. The less nonzero elements in E_1 are, the lower γ_1 is. The analysis of γ_1 can be partitioned into two cases in terms of the position of λ's in rows of $\mathbf{H}$.

Case I: We consider E_1 composed of rows with λ's in the same position. In this case, E_1 can be represented as

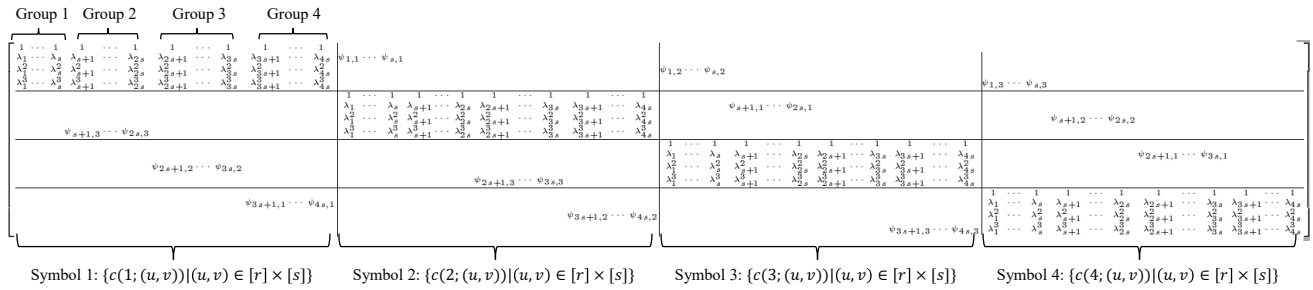

Fig. 1. The parity-check matrix after row and column permutations.

$\sum_{i=0}^{3} l_i h_{i,\hat{u}}$, where $\hat{u} \in [4]\backslash\{u^*\}$ and $l_0, \ldots, l_3 \in \mathbb{B}$ are not all zero. Linear constraints defined by E_1 is

$$\underbrace{\sum_{(u,v)\in[r]\times[s]} (\sum_{j=0}^{3} l_j \lambda_{(u-1)s+v}^{j}) \cdot c(\hat{u};(u,v))}_{(a)} + \underbrace{\sum_{p=1}^{3}\sum_{v\in[s]} l_p \psi_{(\hat{u}-1)s+v,p} \cdot c(\overline{\hat{u}+p};(\hat{u},v))}_{(b)} = 0. \tag{6}$$

If coefficients $\sum_{j=0}^{3} l_j \lambda_{(u-1)s+v}^{j}$ and $l_p \psi_{(\hat{u}-1)s+v,p}$ are nonzero, symbols $c(\hat{u};(u,v))$ and $c(\overline{\hat{u}+p};(\hat{u},v))$ are involved in the repair, respectively. Since symbols in the set $U_{nb}\backslash U_b$ have been downloaded, it is unnecessary to consider their coefficients. In order to achieve a lower γ_1, as more zero coefficients as possible are desired. The following we analyze the minimum number of nonzero coefficients in E_1.

For the part (a) in Eq. (6), if given a coefficient $\sum_{j=0}^{3} l_j \lambda_{i'}^{j} = 0$, i.e., $l_0 = -\sum_{j=1}^{3} l_j \lambda_{i'}^{j}$, where $i' \in [n]\backslash\{(u^*-1)s+1, \ldots, u^*s\}$, we need to analyze that how many zero coefficients can exist. We have the coefficient

$$\sum_{j=0}^{3} l_j \lambda_i^j = -\sum_{j=1}^{3} l_j \lambda_{i'}^{j} + \sum_{j=1}^{3} l_j \lambda_i^j = \sum_{j=1}^{3} l_j (\lambda_i^j - \lambda_{i'}^j). \tag{7}$$

There exist at most three different λ_i's, s.t. Eq. (7)$= 0$, where $i \in [n]\backslash\{(u^*-1)s+1, \ldots, u^*s\}$. Suppose that these three different λ_i's are $\{\lambda_{i_1}, \lambda_{i_2}, \lambda_{i_3}\}$, where $i_1, i_2, i_3 \in [n]\backslash\{(u^*-1)s+1, \ldots, u^*s\}$. If one of $\{i_1, i_2, i_3\}$ equals $(u'-1)s+v'$, i.e., it corresponds to the busy node, the part (a) does not introduce unknown symbols (in the busy node) and will lead to additionally download $3s-3$ symbols. Otherwise, the part (a) introduces one unknown symbol $c(\hat{u};(u',v'))$ and will lead to additionally download $3s-4$ symbols.

For the part (b) in Eq. (6), suppose that for every $p \in [3]\backslash\{p'\}$, $l_p = 0$, where $\overline{\hat{u}+p'} = u^*$. Symbols $\{c(\overline{\hat{u}+p};(\hat{u},v))|p \in [3]\backslash\{p'\}, v \in [s]\}$ are unnecessary to be additionally downloaded. In this condition,

$$E_1 = l_0 h_{0,\hat{u}} + l_{p'} h_{p',\hat{u}},$$

where l_0 and $l_{p'}$ are nonzero. Eq. (6) can be represented as

$$\sum_{(u,v)\in[r]\times[s]} (l_0 + l_{p'} \lambda_{(u-1)s+v}^{p'}) \cdot c(\hat{u};(u,v)) = 0. \tag{8}$$

Thus, in consideration of parts (a) and (b),

1) if E_1 introduces no unknown symbols in the busy node, E_1 together with Eqs. (3), (4) can be viewed as one busy-node repair scheme, i.e., $m=1$. We have $\gamma = \gamma_1 = 3s-3$.

2) if E_1 introduces one unknown symbol in the busy node, we have $\gamma_1 \geq 3s-4$. Eqs. (3), (4) together with E_1 indicate that there are $r+2$ unknown symbols (i.e., r symbols to be repaired and 2 unavailable symbols in the busy node) and $r+1$ equations. Thus, at least E_2 is required to form the busy-node repair scheme, i.e., $m \geq 2$.

Since $3s-4 < 3s-3$, it is necessary to present γ_1 in two conditions for the analysis of γ.

Case II: We consider E_1 composed of rows with λ's in different positions. Without loss of generality, we set $u^* = 1$. The analysis of E_1 can be partitioned into two conditions.

(a) E_1 can be represented as $E_1 = \sum_{u\in Q} \sum_{j=0}^{3} l_{j,u} \cdot h_{j,u}$, where for $\forall u \in Q$, $0 \leq j \leq 3$, $l_{j,u} \in \mathbb{B}$, Q is a subset of $\{2,3,4\}$ and $|Q| = 2$. E_1 will lead to additionally download at least $2\times(2s-1-3) = 4s-8$ symbols. Thus, $\gamma_1 \geq 4s-8 > 3s-3$ in this condition.

(b) E_1 can be represented as $E_1 = \sum_{u\in\{2,3,4\}} \sum_{j=0}^{3} l_{j,u} \cdot h_{j,u}$, where for $\forall u \in \{2,3,4\}$, $0 \leq j \leq 3$, $l_{j,u} \in \mathbb{B}$. E_1 will lead to additionally download at least $3\times(s-3)-1 = 3s-10$ symbols. If $\gamma_1 \geq 3s-9$ holds, that means that the following two properties should be satisfied.

i) There exist $\{i_1, i_2, i_3\} \in \{s+1, \ldots, 2s\}$, $\{i_4, i_5, i_6\} \in \{2s+1, \ldots, 3s\}$ and $\{i_7, i_8, i_9\} \in \{3s+1, \ldots, 4s\}$ such that for $\forall t \in [3]$, $\sum_{j=0}^{3} l_{j,2} \lambda_{i_t}^{j} = 0, \sum_{j=0}^{3} l_{j,3} \lambda_{i_{t+3}}^{j} = 0, \sum_{j=0}^{3} l_{j,4} \lambda_{i_{t+6}}^{j} = 0$.

ii) When $s+1 \leq i \leq 2s$, $\sum_{j=0}^{3} l_{j,3} \lambda_i^j + l_{1,2}\psi_{i,1} = 0$ and $\sum_{j=0}^{3} l_{j,4} \lambda_i^j + l_{2,2}\psi_{i,2} = 0$. When $2s+1 \leq i \leq 3s$, $\sum_{j=0}^{3} l_{j,2} \lambda_i^j + l_{3,3}\psi_{i,3} = 0$ and $\sum_{j=0}^{3} l_{j,4} \lambda_i^j + l_{1,3}\psi_{i,1} = 0$. When $3s+1 \leq i \leq 4s$, $\sum_{j=0}^{3} l_{j,2} \lambda_i^j + l_{2,4}\psi_{i,2} = 0$ and $\sum_{j=0}^{3} l_{j,3} \lambda_i^j + l_{3,4}\psi_{i,3} = 0$.

If $(u'-1)s+v' \in \{i_1, \ldots, i_9\}$, i.e., E_1 introduces no unknown symbol, Eqs. (3), (4) together with E_1 can form a busy-node repair scheme. If $(u'-1)s+v' \notin \{i_1, \ldots, i_9\}$, i.e., E_1 introduces one unknown symbol $c(u';(u',v'))$, we have $\gamma_1 = 3s-10$. Eqs. (3), (4) together with E_1 indicate that there are $r+2$ unknown symbols and $r+1$ equations. Thus, at least E_2 is required to form the busy-node repair scheme. Moreover, since vectors, which are linearly dependent of E_1, have no contribution to the repair, such vectors are viewed as the same as E_1. Considering that if the code is well designed,

a vector E_1 with coefficients $\{l_{j,u}|0 \le j \le 3, u \in \{2,3,4\}\}$ is unique such that $\gamma_1 \ge 3s-9$ holds.

The following we analyze a lower bound of γ.

1) If E_1 satisfies Case I,

1.1) If E_1 introduces no unknown symbols, $\gamma_1 \ge 3s-3$. Thus, we have $m=1$ and $\gamma \ge \gamma_1 \ge 3s-3$.

1.2) If E_1 introduces one unknown symbol, $\gamma_1 \ge 3s-4$. At least E_2 is required. If E_2 satisfies Case I, E_2 must be composed of rows with λ's in the same position as E_1 such that additional bandwidth $\gamma_{\{1,2\}}$ brought by $\{E_1, E_2\}$ will be the least. According to the above analysis, $\gamma_{\{1,2\}} = \gamma_1$, if and only if $E_2 = a \cdot E_1$, where $a \in \mathbb{B}$. However, if $E_2 = a \cdot E_1$, E_2 has no contribution to recover the failed node. Therefore, if E_2 contributes to the busy-node node repair, $\gamma_{\{1,2\}} > \gamma_1 \ge 3s-4$ thus to $\gamma_{\{1,2\}} \ge 3s-3$. Thus, we have $\gamma \ge \gamma_{\{1,2\}} \ge 3s-3$. If E_2 satisfies Case II, we have $\gamma_{\{1,2\}} \ge 3s-4+2(s-3) = 5s-10$ thus to $\gamma > 3s-3$.

2) If E_1 satisfies Case II.(a), $\gamma \ge \gamma_1 > 3s-3$;

3) If E_1 satisfies Case II.(b), when (u', v') does not take one of 9 specific tuples, E_1 introduces one unknown symbol. At least E_2 is required. Since $E_2 \ne a \cdot E_1$, we have $\gamma_{\{1,2\}} \ge 3s-10+2s = 5s-10$, where $a \in \mathbb{B}$. Thus, we can obtain that $\gamma > \frac{9\times(3s-9)+(3s-9)\times(5s-10)}{3s} = 5s-16+\frac{3}{s} > 3s-3$ in terms of the average bandwidth.

To sum up, we have $\gamma \ge 3s-3$. Thus, a lower bound of busy-node repair is $\gamma_{nb} - 1 + \gamma \ge (2n-s-r)-1+3s-3 = 10s-8$ when failed node and busy node are in different groups.

B. Failed node and busy node are in the same group

This subsection considers the case that the failed node and busy node are in the same group, i.e., $u' = u^*$. The set of symbols downloaded in Eqs. (3) and (4) is $U_{nb} \backslash U_b$. Eqs. (3) and (4) turn to be $r+4$ unknown symbols, $U_f \cup U_b$, and r equations. Thus, at least another four equations are required. We first present a lower bound of E_i, where $1 \le i \le m$. Then, the properties that additional vectors should satisfy are analyzed. Last, a lower bound of busy-node repair bandwidth is derived from going through all cases.

Continue the settings in Section III.A. In this case, $m=4$. The analysis of lower bound when the failed node and busy node are in the same group is similar to that of lower bound when failed node and busy node are in different groups. The difference is due to the position of the busy node. Any additional equation would not introduce the unknown symbol in the busy node. Without loss of generality, we set $u' = u^* = 1$ in the following analysis.

Similar to the analysis in Section III.A, we can obtain a lower bound of E_i drawn as below, where $1 \le i \le 4$.

Case I: Consider E_i composed of rows with λ's in the same position. We have $\gamma_i \ge 3s-3$.

Case II: Consider E_i composed of rows with λ's in different positions.

(a) E_i can be represented as

$$E_i = \sum_{u \in Q} \sum_{j=0}^{3} l_{i,j,u} \cdot h_{j,u}, \tag{9}$$

$$\left[\begin{array}{cccc|cccc} 1 & & & & 1 & & & \\ & \psi_{t^*,1} & & & \lambda_{t'}-\lambda_{t^*} & \psi_{t',1} & & \\ & & \psi_{t^*,2} & & \lambda^2_{t'}-\lambda^2_{t^*} & & \psi_{t',2} & \\ & & & \psi_{t^*,3} & \lambda^3_{t'}-\lambda^3_{t^*} & & & \psi_{t',3} \\ \hline 0 & \sum_{j=0}^3 l_{1,j,2}\lambda_{t^*}^j & \sum_{j=0}^3 l_{1,j,3}\lambda_{t^*}^j & 0 & 0 & \sum_{j=0}^3 l_{1,j,2}\lambda_{t'}^j & \sum_{j=0}^3 l_{1,j,3}\lambda_{t'}^j & 0 \\ 0 & \sum_{j=0}^3 l_{2,j,2}\lambda_{t^*}^j & \sum_{j=0}^3 l_{2,j,3}\lambda_{t^*}^j & 0 & 0 & \sum_{j=0}^3 l_{2,j,2}\lambda_{t'}^j & \sum_{j=0}^3 l_{2,j,3}\lambda_{t'}^j & 0 \\ 0 & \sum_{j=0}^3 l_{3,j,2}\lambda_{t^*}^j & \sum_{j=0}^3 l_{3,j,3}\lambda_{t^*}^j & 0 & 0 & \sum_{j=0}^3 l_{3,j,2}\lambda_{t'}^j & \sum_{j=0}^3 l_{3,j,3}\lambda_{t'}^j & 0 \\ 0 & \sum_{j=0}^3 l_{4,j,2}\lambda_{t^*}^j & \sum_{j=0}^3 l_{4,j,3}\lambda_{t^*}^j & 0 & 0 & \sum_{j=0}^3 l_{4,j,2}\lambda_{t'}^j & \sum_{j=0}^3 l_{4,j,3}\lambda_{t'}^j & 0 \end{array}\right]$$

Fig. 2. The matrix $\mathbf{H}_{rp}$ for the proof of Property2.

where for $\forall u \in Q$, $0 \le j \le 3$, $l_{i,j,u} \in \mathbb{B}$, Q is a subset of $\{2,3,4\}$ and $|Q| = 2$. We have $\gamma_i \ge 4s-6$.

(b) E_i can be represented as $E_i = \sum_{u \in \{2,3,4\}} \sum_{j=0}^{3} l_{i,j,u} \cdot h_{j,u}$, where for $\forall u \in \{2,3,4\}$, $0 \le j \le 3$, $l_{i,j,u} \in \mathbb{B}$. We have $\gamma_i \ge 3s-9$.

Then, we analyze the necessary properties that $\{E_1, \ldots, E_4\}$ should satisfy, if additional equations defined by $\{E_1, \ldots, E_4\}$ together with Eqs. (3), (4) can recover the failed node. According to Eqs. (3), (4) and additional equations defined by vectors $\{E_1, \ldots, E_4\}$, the repair process can be written as $\mathbf{H}_{rp} \cdot \left[\begin{array}{cc} \mathbf{c}_{(u^*,v^*)} & \mathbf{c}_{(u',v')} \end{array}\right]^{\mathrm{T}} = \mathbf{H}_{dl} \cdot \hat{\mathbf{c}}^{\mathrm{T}}$, where $\mathbf{H}_{rp}$ and $\mathbf{H}_{dl}$ are coefficient matrices and $\hat{\mathbf{c}}$ denotes the vector composed of downloaded symbols. When the coefficient matrix $\mathbf{H}_{rp}$ is full rank, symbols in the failed node can be recovered. Employing row transformations on $\mathbf{H}_{rp}$, we have

$$\left[\begin{array}{cccc} 1 & & & 1 \\ & \psi_{t^*,1} & & \lambda_{t'}-\lambda_{t^*} \quad \psi_{t',1} \\ & & \psi_{t^*,2} & \lambda^2_{t'}-\lambda^2_{t^*} \quad \psi_{t',2} \\ & & \psi_{t^*,3} & \lambda^3_{t'}-\lambda^3_{t^*} \quad \psi_{t',3} \\ \hline & & & F_1 \\ & & & \vdots \\ & & & F_4 \end{array}\right],$$

where $t^* = (u^*-1)s+v^*$, $t' = (u'-1)s+v'$ and $F_1, \ldots, F_4$ correspond to $E_1, \ldots, E_4$, respectively.

Property 1. *There at most exist two of $E_1, \ldots, E_4$ composed of rows with λ's in the same position.*

Proof. Assume that there exist no less than two of $E_1, \ldots, E_4$ composed of rows with λ's in the same position. We might as well set E_i being represented as $\sum_{j=0}^{3} l_{i,j} h_{j,2}$, where $l_{i,0}, \ldots, l_{i,3} \in \mathbb{B}$ are not all zero and $i = 1,2,3$. The matrix

$$\left[\begin{array}{c} F_1 \\ F_2 \\ F_3 \end{array}\right] = \left[\begin{array}{cccc|cccc} 0 & \sum_{j=0}^3 l_{1,j}\lambda_{t^*}^j & 0 & 0 & 0 & \sum_{j=0}^3 l_{1,j}\lambda_{t'}^j & 0 & 0 \\ 0 & \sum_{j=0}^3 l_{2,j}\lambda_{t^*}^j & 0 & 0 & 0 & \sum_{j=0}^3 l_{2,j}\lambda_{t'}^j & 0 & 0 \\ 0 & \sum_{j=0}^3 l_{3,j}\lambda_{t^*}^j & 0 & 0 & 0 & \sum_{j=0}^3 l_{3,j}\lambda_{t'}^j & 0 & 0 \end{array}\right]$$

is not full rank. □

Property 2. *If $E_1, \ldots, E_4$ are all composed of rows represented as Eq.* (9) *with the same Q, the matrix $\mathbf{H}_{rp}$ is not full rank. That is to say, the necessary condition that the matrix $\mathbf{H}_{rp}$ is full rank is that rows with λ's in three different positions should all be involved in compositions of $\{E_1, \ldots, E_4\}$.*

Proof. We might as well set $Q = \{2,3\}$. Vectors $E_1, \ldots, E_4$ can be represented as Eq. (9). The matrix $\mathbf{H}_{rp}$ presented in Fig. (2) is not full rank. □

Compositions of $E_1, \ldots, E_4$ can be discussed in terms of the above two properties. Then, in consideration of a lower bound of E_i, it can be verified that a lower bound of γ is $7s-8$, where $1 \le i \le 4$. Thus, a lower bound of the busy-node repair is $\gamma_{nb} - 4 + \gamma = 7s-4-4+7s-8 = 14s-16$ when the failed node and busy node are in the same group.

IV. Optimal Busy-Node Repair Scheme When Failed Node and Busy Node Are In Different Groups

In this section, we propose a busy-node repair scheme when the failed node and the busy node are in different groups. The proposed scheme shows that it requires a cubic constraint among elements in the parity-check matrix to achieve a lower bound analyzed in Section III.A.

A. Optimal busy-node repair scheme

Based on the analysis in Section III.A, we propose to design E_1 as $E_1 = -\lambda^3_{(u'-1)s+v'} h_{0,\hat{u}} + h_{3,\hat{u}}$, where $\overline{3+\hat{u}} = u^*$. The linear constraint defined by the vector E_1 is

$$\begin{aligned} &\sum_{(u,v)\in[r]\times[s]} (\lambda^3_{(u-1)s+v} - \lambda^3_{(u'-1)s+v'}) \cdot c(\hat{u};(u,v)) \\ &+ \sum_{v\in[s]} \psi_{(\hat{u}-1)s+v,3} \cdot c(u^*;(\hat{u},v)) = 0. \end{aligned} \tag{10}$$

The busy-node repair is composed of Eqs. (3), (4) and (10).

Then we show that how can the proposed scheme achieve a lower bound. Since $(\lambda^3_{(u'-1)s+v'} - \lambda^3_{(u'-1)s+v'}) \cdot c(\hat{u};(u',v')) = 0$, the additional linear constraint (10) does not introduce new unknown symbol. Eqs. (3), (4) and (10) indicate that there are $r+1$ unknown symbols and $r+1$ equations. Symbols $\{c(u^*;(\hat{u},v))|v\in[s]\}$ have been downloaded when there is no busy node, so they do not increase the repair bandwidth. Thus, the analysis of the busy-node repair bandwidth only needs to consider the items $(\lambda^3_{(u-1)s+v} - \lambda^3_{(u'-1)s+v'}) \cdot c(\hat{u};(u,v))$, where $(u,v)\in[r]\times[s]$.

Lemma 1. *Suppose that i_1 is a constant value in the finite field $\mathbb{B}$ of size $|\mathbb{B}|$. If $3|(|\mathbb{B}|-1)$, there exist three different $i\in\mathbb{B}$ such that $i^3 - i_1^3 = 0$.*

Proof. Let α denote the primitive element of $\mathbb{B}$. Any value in $\mathbb{B}$ can be written as the power of α. Let $i_1 = \alpha^{a_1}$ and $i = \alpha^a$, where $0 \le a_1, a \le |\mathbb{B}|-2$. If $i^3 - i_1^3 = 0$, $i^3 = i_1^3$ or $i_1^3 \cdot \alpha^{|\mathbb{B}|-1}$ or $i_1^3 \cdot \alpha^{|\mathbb{B}|-1} \cdot \alpha^{|\mathbb{B}|-1}$, i.e., $(\alpha^a)^3 = (\alpha^{a_1})^3$ or $(\alpha^{a_1})^3 \cdot \alpha^{|\mathbb{B}|-1}$ or $(\alpha^{a_1})^3 \cdot \alpha^{|\mathbb{B}|-1} \cdot \alpha^{|\mathbb{B}|-1}$, because $3a \le 3(|\mathbb{B}|-2)$. Thus, $3a = 3a_1$ or $3a_1+(|\mathbb{B}|-1)$ or $3a_1+2(|\mathbb{B}|-1)$. If $3|(|\mathbb{B}|-1)$, a can take three different values $a_1, (a_1+\frac{|\mathbb{B}|-1}{3}) \bmod (|\mathbb{B}|-1), (a_1+\frac{2(|\mathbb{B}|-1)}{3}) \bmod (|\mathbb{B}|-1)$, i.e., i can take three different values. □

According to Lemma 1, there exist at most three different $\lambda_{(u-1)s+v}$ such that $(\lambda^3_{(u-1)s+v} - \lambda^3_{(u'-1)s+v'}) = 0$ which is termed as *cubic constraint*. Symbols required to be downloaded additionally are at least $\gamma_c = (r-1)s - 3 = n - s - 3$, i.e., the bandwidth of the proposed scheme is $10s-8$, if the code is well designed to reach it.

B. A necessary and sufficient condition of our repair

According to Eqs. (3), (4) and (10), the repair process can be written as $\mathbf{H}_{rp} \cdot \left[\begin{array}{cc} \mathbf{c}_{(u^*,v^*)} & c(u^*;(u',v')) \end{array}\right]^{\mathrm{T}} = \mathbf{H}_{dl} \cdot \hat{\mathbf{c}}^{\mathrm{T}}$, where $\mathbf{H}_{rp}$ and $\mathbf{H}_{dl}$ are coefficient matrices and $\hat{\mathbf{c}}$ denotes the vector composed of downloaded symbols. The coefficient matrix $\mathbf{H}_{rp}$ being full rank is the necessary and sufficient condition that the proposed repair scheme can recover the failed node.

Employing column permutation on $\mathbf{H}_{rp}$, we have

$$\left[\begin{array}{cccc|c} 1 & 0 & 0 & 0 & 1 \\ \lambda_{t^*} & \psi_{t^*,1} & 0 & 0 & \lambda_{t'} \\ \lambda^2_{t^*} & 0 & \psi_{t^*,2} & 0 & \lambda^2_{t'} \\ \lambda^3_{t^*} & 0 & 0 & \psi_{t^*,3} & \lambda^3_{t'} \\ 0 & \lambda^3_{t^*} - \lambda^3_{t'} & 0 & 0 & \psi_{t',3} \end{array}\right] \text{ or } \left[\begin{array}{cccc|c} 1 & 0 & 0 & 0 & 1 \\ \lambda_{t^*} & \psi_{t^*,1} & 0 & 0 & \lambda_{t'} \\ \lambda^2_{t^*} & 0 & \psi_{t^*,2} & 0 & \lambda^2_{t'} \\ \lambda^3_{t^*} & 0 & 0 & \psi_{t^*,3} & \lambda^3_{t'} \\ 0 & \lambda^3_{t^*} - \lambda^3_{t'} & 0 & 0 & 0 \end{array}\right],$$

where $t^* = (u^*-1)s+v^*$ and $t' = (u'-1)s+v'$. $\mathbf{H}_{rp}$ being full rank means that

$$\left|\begin{array}{cc} \psi_{t^*,1} & \lambda_{t'} - \lambda_{t^*} \\ \lambda^3_{t^*} - \lambda^3_{t'} & \psi_{t',3} \end{array}\right| \neq 0 \text{ and } \lambda^3_{t^*} - \lambda^3_{t'} \neq 0. \tag{11}$$

To sum up, the proposed scheme can achieve a lower bound of bandwidth by linear combinations the row of the parity-check matrix. In order to achieve a lower bound, the code design should satisfy conditions drawn as below. Given the failed node with the tuple (u^*,v^*), the busy node with the tuple (u',v') and $\lambda_{(u'-1)s+v'}$, where $u' \neq u^*$,

1) the size of the finite field $\mathbb{B}$ should satisfy that $3|(\mathbb{B}-1)$.
2) elements in the parity-check matrix should satisfy cubic constraints. Select two different tuples $(u_1,v_1),(u_2,v_2)$ and let $\lambda_{(u_1-1)s+v_1} = \lambda_{(u'-1)s+v'} \cdot \alpha^{\frac{|\mathbb{B}|-1}{3}}$ and $\lambda_{(u_2-1)s+v_2} = \lambda_{(u'-1)s+v'} \cdot \alpha^{\frac{2(|\mathbb{B}|-1)}{3}}$, where $(u_1,v_1),(u_2,v_2) \in [r]\times[s]$, $u_1,u_2 \neq u^*$, $(u_1,v_1),(u_2,v_2) \neq (u',v')$ and α is the primitive element in $\mathbb{B}$.
3) the selection of $\lambda_{(u^*-1)s+v^*}, \psi_{(u^*-1)s+v^*,1}, \psi_{(u'-1)s+v',3}$ should satisfy Eq. (11).

Example 1. *Suppose that a $(12,8,4)$ MDS-SSL code with the parity check matrix $\mathbf{H}$ satisfies $\lambda_1^3 = \lambda_2^3 = \lambda_3^3$, $\lambda_4^3 = \lambda_5^3 = \lambda_6^3$, $\lambda_7^3 = \lambda_8^3 = \lambda_9^3$ and $\lambda_{10}^3 = \lambda_{11}^3 = \lambda_{12}^3$. If the node $(1,1)$ fails and the node $(2,1)$ is busy, we recover the failed node by vectors $h_{0,1}, h_{1,1}, h_{2,1}, h_{3,1}$ and $-\lambda_4^3 h_{0,2} + h_{3,2}$. Downloaded symbols are $\mathbf{c}_{(1,2)}, \mathbf{c}_{(1,3)}, c(1;(2,2)), c(1;(2,3))$ and $\{c(x;(u,v))|x\in[2], u\in\{3,4\}, v\in[3]\}$. The number of downloaded symbols is 22, i.e., busy-node repair bandwidth is 22, which achieves the bound analyzed.*

References

[1] A. Cidon, S. Rumble, R. Stutsman, S. Katti, J. Ousterhout, and M. Rosenblum, "Copysets: Reducing the frequency of data loss in cloud storage," in *Proc. 2013 USENIX Annual Technical Conference (USENIX ATC 13)*, San Jose, CA, Jun. 2013, pp. 37–48, USENIX Association.

[2] A. S. Rawat, I. Tamo, V. Guruswami, and K. Efremenko, "MDS code constructions with small sub-packetization and near-optimal repair bandwidth," *IEEE Trans. Inf. Theory*, vol. 64, no. 10, pp. 6506–6525, Feb. 2018.

[3] A. G. Dimakis, P. B. Godfrey, Y. Wu, M. J. Wainwright, and K. Ramchandran, "Network coding for distributed storage systems," *IEEE Trans. Inf. Theory*, vol. 56, no. 9, pp. 4539–4551, Sept. 2010.

[4] V. R. Cadambe, S. A. Jafar, H. Maleki, K. Ramchandran, and C. Suh, "Asymptotic interference alignment for optimal repair of MDS codes in distributed storage," *IEEE Trans. Inf. Theory*, vol. 59, no. 5, pp. 2974–2987, May. 2013.

[5] B. Sasidharan, G. K. Agarwal, and P. V. Kumar, "A high-rate MSR code with polynomial sub-packetization level," in *Proc. 2015 IEEE International Symposium on Information Theory (ISIT)*.

[6] M. Ye and A. Barg, "Explicit constructions of high-rate MDS array codes with optimal repair bandwidth," *IEEE Trans. Inf. Theory*, vol. 63, no. 4, pp. 2001–2014, Apr. 2017.

[7] J. Li and X. Tang, "Systematic construction of MDS codes with small sub-packetization level and near optimal repair bandwidth," in *Proc. 2019 IEEE International Symposium on Information Theory (ISIT)*, 2019, pp. 1067–1071.

[8] K. W. Shum and Y. Hu, "Cooperative regenerating codes," *IEEE Trans. Inf. Theory*, vol. 59, no. 11, pp. 7229–7258, Jul. 2013.