A Class of MSR Codes for Clustered Distributed Storage

Jy-yong Sohn, Beongjun Choi and Jaekyun Moon
KAIST
School of Electrical Engineering
Email: {jysohn1108, bbzang10}@kaist.ac.kr, jmoon@kaist.edu

***Abstract*—Clustered distributed storage models real data centers where intra- and cross-cluster repair bandwidths are different. In this paper, exact-repair minimum-storage-regenerating (MSR) codes achieving capacity of clustered distributed storage are designed. Focus is given on two cases: $\epsilon = 0$ and $\epsilon = 1/(n-k)$, where ϵ is the ratio of the available cross- and intra-cluster repair bandwidths, n is the total number of distributed nodes and k is the number of contact nodes in data retrieval. The former represents the scenario where cross-cluster communication is not allowed, while the latter corresponds to the case of minimum cross-cluster bandwidth allowing minimum storage overhead. For the $\epsilon = 0$ case, two types of locally repairable codes are proven to achieve the MSR point. As for $\epsilon = 1/(n-k)$, MDS codes achieve the MSR points for $n = Lk$, where L is the number of clusters.**

I. Introduction

Distributed storage systems (DSSs) have been deployed by various enterprises to reliably store massive amounts of data under frequent storage node failure events. A failed node is regenerated (repaired) by collecting information from other survived nodes with the regeneration process guided by a predefined network coding scheme. Under this setting, Dimakis *et al.* [1] obtained an expression for the maximum reliably storable file size, denoted by *capacity* $\mathcal{C}(\alpha, \gamma)$, as a function of given system parameters: the node capacity α and the bandwidth γ required for repairing a failed node. The capacity analysis in [1] underscores the following key messages. First, there exists a network coding scheme which utilizes the (α, γ) resources and enables a reliable storage of a file of size $\mathcal{C}(\alpha, \gamma)$. Second, it is not feasible to find a network coding scheme which can reliably store a file larger than $\mathcal{C}(\alpha, \gamma)$, given the available resources of (α, γ). In subsequent research efforts, the authors of [2]–[6] proposed explicit network coding schemes which achieve the capacity of DSSs. These coding schemes are optimal in the sense of efficiently utilizing (α, γ) resources for maintaining reliable storage.

The focus on the clustered nature of distributed storage has been a recent research direction taken by several researchers [7]–[10]. According to these recent papers, storage nodes dispersed into multiple *racks* in real data centers are seen as forming *clusters*. In particular, the authors of the present paper proposed a system model for clustered DSSs in [7] that reflects the difference between intra- and cross-cluster bandwidths. In the system model of [7], the file to be stored is coded and distributed into n storage nodes, which are evenly dispersed into L clusters. Each node has storage capacity of α, and the data collector contacts arbitrary k out of n existing nodes to retrieve the file. Since nodes are dispersed into multiple clusters, the regeneration process involves utilization of both intra- and cross-cluster repair bandwidths, denoted by β_I and β_c, respectively. In this proposed system model, the authors of [7] obtained a closed-form expression for the maximum reliably storable file size, or *capacity* $\mathcal{C}(\alpha, \beta_I, \beta_c)$, of the clustered DSS. Furthermore, it has been shown that network coding exists that can achieve the capacity of clustered DSSs. However, explicit constructions of capacity-achieving network coding schemes for clustered DSSs have yet to be found.

This paper proposes a network coding scheme which achieves capacity of the clustered DSS, with a minimum required node storage overhead. In other words, the suggested code is shown to be a minimum-storage-regenerating (MSR) code of the clustered DSS. This paper focuses on two important cases of $\epsilon = 0$ and $\epsilon = 1/(n-k)$, where $\epsilon := \beta_c/\beta_I$ represents the ratio of cross- to intra-cluster repair bandwidths. The former represents the system where cross-cluster communication is not possible. The latter corresponds to the minimum ϵ value that can achieve the minimum storage overhead of $\alpha = \mathcal{M}/k$, where $\mathcal{M}$ is the file size. When $\epsilon = 0$, it is shown that appropriate application of locally repairable codes suggested in [11], [12] achieves the MSR point for general n, k, L settings with the application rule depending on the parameter setting. For the $\epsilon = 1/(n-k)$ case, an explicit coding scheme utilizing a plain MDS code is suggested which is proven to be an MSR code under the conditions of $n = Lk$. There have been some previous works [9], [10], [13], [14] on code construction for DSS with clustered storage nodes, but to a limited extent. The works of [10], [13] suggested a coding scheme which can reduce the cross-cluster repair bandwidth, but these schemes are not proven to be an MSR code that achieves capacity of clustered DSSs with minimum storage overhead. The authors of [14] provided an explicit coding scheme which reduces the repair bandwidth of a clustered DSS under the condition that each failed node can be exactly regenerated by contacting any one of other clusters. However, the approach of [14] is different from that of the present paper in the sense that it does not consider the scenario with unequal intra- and cross-cluster repair bandwidths. Moreover, the coding scheme proposed in [14] is shown to be a minimum-bandwidth-regenerating (MBR) code for some limited parameter setting, while the present paper deals with an MSR code. An MSR code for clustered DSSs has been suggested in [9], but this paper has the data retrieval condition different from the present paper. The authors of [9] considered the scenario where data can be collected by contacting arbitrary k out of n clusters, while data can be retrieved by contacting arbitrary k out of n nodes

978-1-5386-4780-6/18/$31.00©2018 IEEE

in the present paper. Thus, the storage versus repair-bandwidth tradeoff curves for the present paper and [9] are different, since data retrieval conditions are different. In short, the code in [9] and the code in this paper achieve different tradeoff curves.

II. Backgrounds and Notations

A given file of $\mathcal{M}$ symbols is encoded and distributed into n nodes, each of which can store α symbols. The storage nodes are evenly distributed into $L \geq 2$ clusters, so that each cluster contains $n_I := n/L$ nodes. A failed node is regenerated by obtaining information from other survived nodes: $n_I - 1$ nodes in the same cluster help by sending β_I symbols each, while $n - n_I$ nodes in other clusters help by sending β_c symbols each. Thus, the overall repair bandwidth is expressed as

$$\gamma = (n_I - 1)\beta_I + (n - n_I)\beta_c. \tag{1}$$

It is assumed that any failed node is repaired by contacting all the remaining nodes, i.e., $n-1$ nodes. This is the capacity-maximizing setting, according to Proposition 1 of [8].

A data collector (DC) retrieves the original file $\mathcal{M}$ by contacting arbitrary k (out of n) nodes - this property is called the maximum-distance-separable (MDS) property. The clustered distributed storage system with parameters n, k, L is called an $[n, k, L]$-clustered DSS. In an $[n, k, L]$-clustered DSS with given parameters of α, β_I, β_c, capacity $\mathcal{C}(\alpha, \beta_I, \beta_c)$ is defined in [8] as the maximum amount of data that can be reliably stored. The closed-form expression for $\mathcal{C}(\alpha, \beta_I, \beta_c)$ is obtained in Theorem 1 of [8]. Aiming at reliably storing file $\mathcal{M}$, the set of $(\alpha, \beta_I, \beta_c)$ values is said to be *feasible* if $\mathcal{C}(\alpha, \beta_I, \beta_c) \geq \mathcal{M}$. Note that for a given $\epsilon = \beta_c/\beta_I$ value, finding the feasible $(\alpha, \beta_I, \beta_c)$ values is equivalent to examining the feasible (α, γ) pair using (1). According to Corollaries 1 and 2 of [8], for all $0 \leq \epsilon \leq 1$, the set of feasible (α, γ) points shows the optimal trade-off relationship between α and γ, as illustrated in Fig. 1. In the optimal trade-off curve, the point with minimum node storage size α is called the minimum-storage-regenerating (MSR) point. Explicit regenerating codes that achieve the MSR point are called the MSR codes. According to Theorem 3 of [8], node capacity of the MSR point satisfies

$$\alpha_{\text{msr}} = \mathcal{M}/k \qquad \text{if } \epsilon \geq \frac{1}{n-k}, \tag{2}$$

$$\alpha_{\text{msr}} > \mathcal{M}/k \qquad \text{if } 0 \leq \epsilon < \frac{1}{n-k}. \tag{3}$$

Note that $\alpha = \mathcal{M}/k$ is the minimum storage overhead to satisfy the MDS property, as stated in [1]. Thus, $\epsilon = 1/(n-k)$ is the scenario with minimum cross-cluster communication when the minimum storage overhead constraint $\alpha = \mathcal{M}/k$ is imposed. Here we introduce some useful notations. For a positive integer n, $[n]$ represents the set $\{1, 2, \cdots, n\}$. For natural numbers a and b, we use the notation $a \mid b$ if a divides b. Similarly, write $a \nmid b$ if a does not divide b. For given k and n_I, we define

$$q := \lfloor \frac{k}{n_I} \rfloor, \tag{4}$$

$$m := mod(k, n_I) = k - qn_I, \tag{5}$$

which represent the quotient and remainder of k/n_I, respectively. As in [8], we assume that $k \geq n_I$ holds throughout the paper.

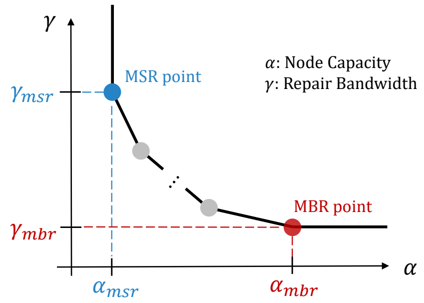


Fig. 1: The optimal trade-off relationship between α and γ in the clustered distributed storage modeled in [8]

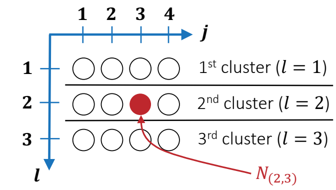


Fig. 2: Two-dimensional representation of clustered distributed storage ($n = 12, L = 3, n_I = n/L = 4$)

For vectors, we use bold-faced lower case letters. For a given vector $\mathbf{a}$, the transpose of $\mathbf{a}$ is denoted as $\mathbf{a}^T$. For natural numbers m and $n \geq m$, the set $\{y_m, y_{m+1}, \cdots, y_n\}$ is represented as $\{y_i\}_{i=m}^n$. For a matrix G, the entry at the i^{th} row and j^{th} column is denoted as $G_{i,j}$. We also express the nodes in a clustered DSS using a two-dimensional representation: in the structure illustrated in Fig. 2, $N(l, j)$ represents the node at the l^{th} row and the j^{th} column. Finally, we recall definitions on the locally repairable codes (LRCs) in [11], [12]. An $(n, k, r)-$LRC [12] represents a code of length n, which is encoded from k information symbols. Every coded symbol of the $(n, k, r)-$LRC can be regenerated by accessing at most r other symbols. An $(n, r, d, \mathcal{M}, \alpha)-$LRC [11] takes a file of size $\mathcal{M}$ and encodes it into n coded symbols, where each symbol is composed of α bits. Moreover, any coded symbol can be regenerated by contacting at most r other symbols, and the code has the minimum distance of d.

III. MSR Code Design for $\epsilon = 0$

In this section, MSR codes for $\epsilon = 0$ (*i.e.*, $\beta_c = 0$) is designed. Under this setting, no cross-cluster communication is allowed in the node repair process. First, the system parameters for the MSR point are examined. Second, locally repairable codes (LRCs) suggested in [11], [12] are proven to achieve the MSR point; the code in [11] is applicable when $n_I \mid k$, while the code in [12] is suitable for general n, k, L values.

A. Parameter Setting for the MSR Point

We consider the MSR point $(\alpha, \gamma) = (\alpha_{\text{msr}}, \gamma_{\text{msr}})$ which can reliably store file $\mathcal{M}$. The following property specifies the system parameters for the $\epsilon = 0$ case.

Proposition 1. *Consider an [n,k,L] clustered DSS to reliably*

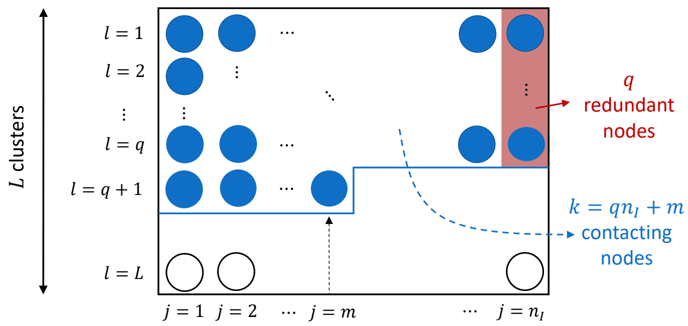


Fig. 3: An example of contacting k nodes with q redundant nodes. This explains why $\alpha = \mathcal{M}/(k-q)$ holds in Proposition 1.

store file $\mathcal{M}$. The MSR point for $\epsilon = 0$ is

$$(\alpha_{msr}, \gamma_{msr}) = \left(\frac{\mathcal{M}}{k-q}, \frac{\mathcal{M}}{k-q}(n_I - 1) \right), \tag{6}$$

where q is defined in (4). This point satisfies $\alpha = \beta_I$.

Proof: The full proof is in [15]. ■

Here we briefly explain the physical meaning of $\alpha = \mathcal{M}/(k-q)$. Consider the scenario of contacting $k = qn_I + m$ nodes as in Fig. 3. Since an arbitrary node needs to be repaired by contacting only the nodes in the same cluster (i.e. $\epsilon = 0$), the content of a node $N(l_0, j_0)$ can be expressed as a function of other nodes $\cup_{j \neq j_0} N(l_0, j)$ in the same cluster. In other words, in the case of contacting the entire nodes in a cluster, there exists at least one redundant node in the cluster which does not provide any additional information. Thus, in the case of contacting k nodes as in Fig. 3, at most $(k-q)\alpha$ symbols are meaningful, which is equal to the file size $\mathcal{M}$ for the MSR point in (6). Since it is required to access the original file by contacting *arbitrary* k nodes (i.e., MDS property), we can verify that $\alpha = \mathcal{M}/(k-q)$ is the minimum node size to enable both the node repair with $\epsilon = 0$ and the MDS property.

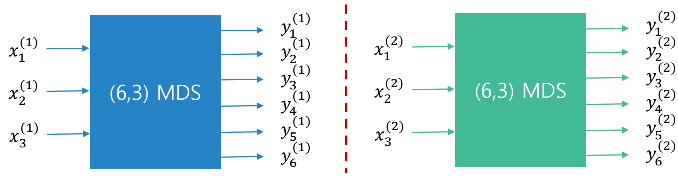
B. Code Construction for $n_I \mid k$

We now examine how to construct an MSR code for the $n_I \mid k$ case. The following theorem shows that a locally repairable code constructed in [11] with locality $r = n_I - 1$ is a valid MSR code for $n_I \mid k$.

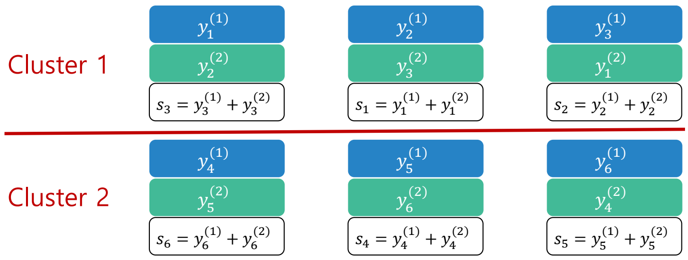
Theorem 1 (Exact-repair MSR Code Construction for $\epsilon = 0, n_I \mid k$) *Let $\mathbb{C}$ be the $(n, r, d, \mathcal{M}, \alpha)-$LRC explicitly constructed in [11] for locality $r = n_I - 1$. Consider allocating coded symbols of $\mathbb{C}$ in a $[n, k, L]-$clustered DSS, where $r+1 = n_I$ nodes within the same repair group of $\mathbb{C}$ are located in the same cluster. Then, the code $\mathbb{C}$ is an MSR code for the $[n, k, L]-$ clustered DSS under the conditions of $\epsilon = 0$ and $n_I \mid k$.*

Proof: The full proof is in [15]. ■

Fig. 4 illustrates an example of the MSR code for the $\epsilon = 0$ and $n_I \mid k$ case, which is constructed using the LRC in [11]. In the $[n, k, L] = [6, 3, 2]$ clustered DSS scenario, the parameters are set to $\alpha = n_I = n/L = 3$ and $\mathcal{M} = (k-q)\alpha = (k - \lfloor k/n_I \rfloor)\alpha = 6$. Thus, each storage node contains $\alpha = 3$ symbols, while the $[n, k, L]$ clustered DSS aims to reliably store a file of size $\mathcal{M} = 6$. This code has two properties, *exact regeneration* and *data reconstruction*: 1) Any failed node can be exactly regenerated by contacting $n_I - 1 = 2$ nodes in the same cluster, and 2) Contacting any $k = 3$ nodes can recover the original file $\{x_i^{(j)} : i \in [3], j \in [2]\}$ of size $\mathcal{M} = 6$.



(a) MDS Precoding



(b) Allocation of coded symbols into n nodes

Fig. 4: MSR code for $\epsilon = 0$ with $n_I \mid k$ $(n = 6, k = 3, L = 2)$. The construction rule follows the instruction in [11], while the concept of the *repair group* in [11] can be interpreted as the *cluster* in the present paper, as stated in Theorem 1.

The first property is obtained from the fact that $y_i^{(1)}, y_i^{(2)}$ and $s_i = y_i^{(1)} + y_i^{(2)}$ form a $(3, 2)$ MDS code for $i \in [6]$. The second property is obtained as follows. For contacting arbitrary $k = 3$ nodes, three distinct coded symbols $\{y_{i_1}^{(1)}, y_{i_2}^{(1)}, y_{i_3}^{(1)}\}$ having superscript one and three distinct coded symbols $\{y_{j_1}^{(2)}, y_{j_2}^{(2)}, y_{j_3}^{(2)}\}$ having superscript two can be obtained for some $i_1, i_2, i_3 \in [6]$ and $j_1, j_2, j_3 \in [6]$. From Fig. 4a, the information $\{y_{i_1}^{(1)}, y_{i_2}^{(1)}, y_{i_3}^{(1)}\}$ suffices to recover $x_1^{(1)}, x_2^{(1)}, x_3^{(1)}$. Similarly, the information $\{y_{j_1}^{(2)}, y_{j_2}^{(2)}, y_{j_3}^{(2)}\}$ suffices to recover $x_1^{(2)}, x_2^{(2)}, x_3^{(2)}$. This completes the proof for the second property. Note that this coding scheme is already suggested by the authors of [11], while the present paper proves that this code also achieves the MSR point of the $[n, k, L]$ clustered DSS, in the case of $\epsilon = 0$ and $n_I \mid k$.

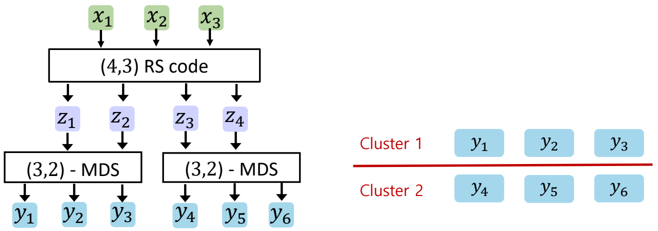
C. Code Construction for arbitrary n, k, L

Here we construct an MSR code which is applicable for arbitrary n, k, L values. The theorem below shows that the optimal $(n, k-q, n_I - 1)-$LRC designed in [12] is a valid MSR code for general n, k, L values, under the setting of $\epsilon = 0$.

Theorem 2 (Exact-repair MSR Code Construction for $\epsilon = 0$) *Let $\mathbb{C}$ be the $(n_0, k_0, r_0)-$LRC constructed in [12] for $n_0 = n, k_0 = k - q$ and $r_0 = n_I - 1$. Consider allocating the coded symbols of $\mathbb{C}$ in a $[n, k, L]-$clustered DSS, where $r + 1 = n_I$ nodes within the same repair group of $\mathbb{C}$ are located in the same cluster. Then, $\mathbb{C}$ is an MSR code for the $[n, k, L]-$clustered DSS under the condition of $\epsilon = 0$.*

Proof: The full proof is in [15]. ■

Fig. 5 illustrates an example of exact-repair MSR code constructed as in Theorem 2. Without losing generality, we



(a) Encoding structure

(b) Allocation of coded symbols into n nodes

Fig. 5: MSR code for $\epsilon = 0$ with $n = 6, k = 4, L = 2$. The encoding structure follows from [12], which constructed $[n_0, k_0, r_0] - LRC$. This paper utilizes $[n, k - q, n_I - 1] - LRC$ to construct MSR code for $[n, k, L]$ clustered DSS, as stated in Theorem 2.

consider the $\alpha = 1$ case; parallel application of this code α times achieves the MSR point for general $\alpha \in \mathbb{N}$, where $\mathbb{N}$ is the set of positivie integers. In the $[n = 6, k = 4, L = 2]$ clustered DSS with $\epsilon = 0$, Proposition 1 implies that $\alpha = 1$, $\mathcal{M} = 3$ and $[n_0, k_0, r_0] = [n, k-q, n_I-1] = [6, 3, 2]$. The code in Fig. 5 satisfies the *exact regeneration* and *data reconstruction* properties: 1) Any failed node can be exactly regenerated by contacting $n_I - 1 = 2$ nodes in the same cluster, and 2) Contacting any $k = 4$ nodes can recover the original file $\{x_i : i \in [3]\}$ of size $\mathcal{M} = 3$. Note that $\{y_i\}_{i=1}^{3}$ in Fig. 5 is a set of coded symbols generated by a $(3, 2)-$MDS code, and this statement also holds for $\{y_i\}_{i=4}^{6}$. This proves the first property. The second property is directly from the result of [12], which states that the minimum distance of the $[n_0, k_0, r_0] - LRC$ is $d = n_0 - k_0 - \lceil k_0/r_0 \rceil + 2 = 6 - 3 - \lceil 3/2 \rceil + 2 = 3$. Note that the $[n_0, k_0, r_0] - LRC$ is already suggested by the authors of [12], while the present paper proves that applying this code with $n_0 = n, k_0 = k - q, r_0 = n_I - 1$ achieves the MSR point of the $[n, k, L]-$clustered DSS with $\epsilon = 0$.

Remark 1. *Theorems 1 and 2 show that LRCs designed in [11], [12] achieve the MSR point for the $\epsilon = 0$ case. Note that various existing $[n, k - q, n_I - 1]-$LRCs, e.g. the code designed in [16], are also possible candidates which achieve the MSR point in a similar way. We leave the detailed analysis on feasible LRCs achieving the MSR point as a future work.*

IV. MSR Code Design for $\epsilon = \frac{1}{n-k}$

We propose an MSR code for $\epsilon = \frac{1}{n-k}$ in clustered DSSs. From (2) and (3), recall that $\frac{1}{n-k}$ is the minimum ϵ value which allows the minimum storage of $\alpha_{\text{msr}} = \mathcal{M}/k$. First, we obtain the system parameters for the MSR point. Secondly, we show that an MDS code achieves the MSR point under the condition $n = Lk$.

A. Parameter Setting for the MSR Point

The following property specifies the system parameters for the $\epsilon = 1/(n - k)$ case. Without a loss of generality, we set the cross-cluster repair bandwidth as $\beta_c = 1$.

Proposition 2. *The MSR point for $\epsilon = 1/(n - k)$ is*

$$(\alpha_{msr}, \gamma_{msr}) = \left(\frac{\mathcal{M}}{k}, \frac{\mathcal{M}}{k} \left(n_I - 1 + \frac{n - n_I}{n - k} \right) \right). \qquad (7)$$

This point satisfies $\alpha = \beta_I = n - k$ and $\mathcal{M} = k(n - k)$.

Proof: The full proof is in [15]. ∎

B. Code Construction for $n = Lk$

The simple MDS code constructed as below is shown to achieve the MSR point for the $\epsilon = 1/(n - k)$ case, when the parameters are such that $n = Lk$.

Theorem 3 (Exact-repair MSR Code Construction for $\epsilon = 1/(n - k), n = Lk$) *Given $\mathcal{M} = k^2(L - 1)$ message symbols, an $[nk(L - 1), k^2(L - 1)]$ MDS code achieves the MSR point for the $[n = Lk, k, L]$ DSS with $\epsilon = 1/(n - k)$, when the coded symbols are equally distributed into $n = LK$ nodes.*

Proof: Suppose that a node in some cluster fails in a $[n = Lk, k, L]$-DSS. In the exact regeneration process, each survived node in the same cluster sends $\beta_I = \alpha = k(L - 1)$ coded symbols, and each survived node in all other clusters sends $\beta_c = \epsilon \beta_I = 1$ coded symbol chosen arbitrarily. The newcomer node obtains

$$\begin{aligned} \gamma &= (n_I - 1)\beta_I + (n - n_I)\beta_c \\ &= (n_I - 1)k(L - 1) + n_I(L - 1) = k^2(L - 1) = \mathcal{M} \end{aligned}$$

coded symbols, where the second last equality is from $n_I = n/L = k$. Using the MDS property, $\mathcal{M}$ message symbols can be obtained in this process. Finally, confirming that the code has system parameters as in (7) completes the proof. ∎

Note that the code constructed in Theorem 3 is based on an MDS code. Here, we provide an example code from a well-known family of MDS codes called *Generalized Reed-Solomon* (GRS) codes. The example code has a generator matrix $G = [I|A]$, where I is the identity matrix and A is a Cauchy matrix. Codes with this form of generator matrix are GRS codes [17].

Consider a $[n = 4, k = 2, L = 2]$ DSS with $\epsilon = 1/(n - k) = 1/2$. Under this setting, an example code is illustrated in Fig. 6 based on a $[8, 4]$-MDS code. The generator matrix is $G = [I_4|A]$, where I_4 is a 4×4 identity matrix and

$$A = \begin{bmatrix} 7 & 2 & 3 & 4 \\ 2 & 7 & 4 & 3 \\ 3 & 4 & 7 & 2 \\ 4 & 3 & 2 & 7 \end{bmatrix} = [a_{ij}] \qquad (8)$$

is a Cauchy matrix using the finite field $GF(2^3)$ based on the primitive polynomial $x^3 + x + 1$. The element $a\alpha^2 + b\alpha + c$ in $GF(2^3)$ is denoted by the decimal number $(abc)_2$, where α is the primitive element. For example, $\alpha + 1$ is denoted by $3 = (011)_2$ in the Cauchy matrix A. The $k^2(L - 1) = 4$ message symbols $\{m_{1,1}, m_{1,2}, m_{2,1}, m_{2,2}\}$ are encoded as

$$\begin{aligned} [m_{1,1}, &m_{1,2}, m_{2,1}, m_{2,2}]G \\ &= [m_{1,1}, m_{1,2}, m_{2,1}, m_{2,2}, p_{1,1}, p_{1,2}, p_{2,1}, p_{2,2}], \qquad (9) \end{aligned}$$

which result in $nk(L - 1) = 8$ coded symbols. According to Proposition 2, the system parameters are

$$\alpha = 2, \mathcal{M} = 4, \beta_I = 2, \beta_c = 1,$$

which holds for the example in Fig. 6. Here we show that the proposed coding scheme satisfies two properties: 1) exact regeneration of any failed node and 2) recovery of $\mathcal{M} = 4$

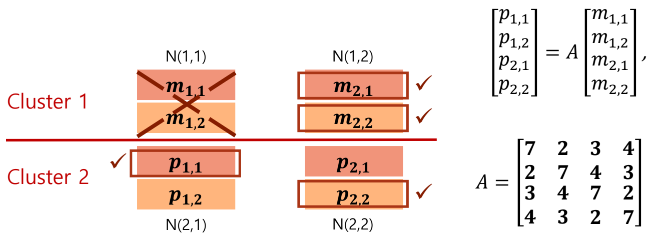


Fig. 6: Repairing a failed node in proposed MSR code example for $n = 4, k = 2, L = 2$

message symbols $\{m_{1,1}, m_{1,2}, m_{2,1}, m_{2,2}\}$ by contacting any $k = 2$ nodes.

1) *Exact regeneration*: Fig. 6 illustrates the regeneration process. Suppose that node $N(1,1)$ containing the message $[m_{1,1}, m_{1,2}]$ fails. Then, node $N(1,2)$ transmits $\beta_I = 2$ symbols, $m_{2,1}$ and $m_{2,2}$. Nodes $N(2,1)$ and $N(2,2)$ transmit $\beta_c = 1$ symbol each, for example, $p_{1,1}$ and $p_{2,2}$, respectively. Then, from the received symbols $m_{2,1}, m_{2,2}, p_{1,1}, p_{2,2}$ and matrix $A = [a_{ij}]$, we obtain

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} := \begin{bmatrix} p_{1,1} - a_{13}m_{2,1} - a_{14}m_{2,2} \\ p_{2,2} - a_{43}m_{2,1} - a_{44}m_{2,2} \end{bmatrix} = \begin{bmatrix} 7 & 2 \\ 4 & 3 \end{bmatrix} \begin{bmatrix} m_{1,1} \\ m_{1,2} \end{bmatrix}.$$

Thus, the contents of the failed node can be regenerated by

$$\begin{bmatrix} m_{1,1} \\ m_{1,2} \end{bmatrix} = \begin{bmatrix} 7 & 2 \\ 4 & 3 \end{bmatrix}^{-1} \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} 3 & 2 \\ 4 & 7 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \end{bmatrix}$$

where the matrix inversion is over $GF(2^3)$. Note that the exact regeneration property holds irrespective of the contents transmitted by $N(2,1)$ and $N(2,2)$, which follows from the MDS property of the utilized $[8,4]$ code.

2) *Data recovery*: Suppose that DC contacts $N(1,1)$ and $N(1,4)$. Then, DC can retrieve message symbols $m_{1,1}, m_{1,2}$ and parity symbols $p_{2,1}, p_{2,2}$. Using the retrieved symbols and the information on the Cauchy matrix $A = [a_{ij}]$, DC additionally obtains

$$\begin{bmatrix} z_1 \\ z_2 \end{bmatrix} := \begin{bmatrix} p_{2,1} - a_{31}m_{1,1} - a_{32}m_{1,2} \\ p_{2,2} - a_{41}m_{1,1} - a_{42}m_{1,2} \end{bmatrix} = \begin{bmatrix} 7 & 2 \\ 2 & 7 \end{bmatrix} \begin{bmatrix} m_{2,1} \\ m_{2,2} \end{bmatrix}.$$

Thus, DC obtains

$$\begin{bmatrix} m_{2,1} \\ m_{2,2} \end{bmatrix} = \begin{bmatrix} 7 & 2 \\ 2 & 7 \end{bmatrix}^{-1} \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = \begin{bmatrix} 1 & 3 \\ 3 & 1 \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \end{bmatrix},$$

which completes the data recovery property.

V. Conclusion

A class of MSR codes for clustered distributed storage modeled in [7] has been constructed. The proposed coding schemes can be applied in practical data centers with multiple racks, where the available cross-rack bandwidth is limited compared to the intra-rack bandwidth. Two important cases of $\epsilon = 0$ and $\epsilon = 1/(n-k)$ are considered, where $\epsilon = \beta_c/\beta_I$ represents the ratio of the available cross- to intra-cluster repair bandwidth. Under the constraint of zero cross-cluster repair bandwidth ($\epsilon = 0$), appropriate application of two locally repairable codes suggested in [11], [12] is shown to achieve the MSR point of clustered distributed storage. Moreover, an explicit MSR coding scheme based on an MDS code is suggested for $\epsilon = 1/(n-k)$, when the system parameters satisfy $n = Lk$.

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. Rashmi, N. B. Shah, P. V. Kumar, and K. Ramchandran, "Explicit construction of optimal exact regenerating codes for distributed storage," in *Communication, Control, and Computing, 2009. Allerton 2009. 47th Annual Allerton Conference on*. IEEE, 2009, pp. 1243–1249.

[3] 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 Transactions on Information Theory*, vol. 59, no. 5, pp. 2974–2987, 2013.

[4] T. Ernvall, "Codes between MBR and MSR points with exact repair property," *IEEE Transactions on Information Theory*, vol. 60, no. 11, pp. 6993–7005, 2014.

[5] S. Goparaju, A. Fazeli, and A. Vardy, "Minimum storage regenerating codes for all parameters," *IEEE Transactions on Information Theory*, vol. 63, no. 10, pp. 6318–6328, 2017.

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

[7] J. y. Sohn, B. Choi, S. W. Yoon, and J. Moon, "Capacity of clustered distributed storage," in *2017 IEEE International Conference on Communications (ICC)*, May 2017.

[8] J. Sohn, B. Choi, S. W. Yoon, and J. Moon, "Capacity of clustered distributed storage," *CoRR*, vol. abs/1710.02821, 2017. [Online]. Available: http://arxiv.org/abs/1710.02821

[9] N. Prakash, V. Abdrashitov, and M. Médard, "The storage vs repair-bandwidth trade-off for clustered storage systems," *arXiv preprint arXiv:1701.04909*, 2017.

[10] Y. Hu, X. Li, M. Zhang, P. P. Lee, X. Zhang, P. Zhou, and D. Feng, "Optimal repair layering for erasure-coded data centers: From theory to practice," *arXiv preprint arXiv:1704.03696*, 2017.

[11] D. S. Papailiopoulos and A. G. Dimakis, "Locally repairable codes," *IEEE Transactions on Information Theory*, vol. 60, no. 10, pp. 5843–5855, 2014.

[12] I. Tamo, D. S. Papailiopoulos, and A. G. Dimakis, "Optimal locally repairable codes and connections to matroid theory," *IEEE Transactions on Information Theory*, vol. 62, no. 12, pp. 6661–6671, 2016.

[13] M. A. Tebbi, T. H. Chan, and C. W. Sung, "A code design framework for multi-rack distributed storage," in *Information Theory Workshop (ITW), 2014 IEEE*. IEEE, 2014, pp. 55–59.

[14] S. Sahraei and M. Gastpar, "Increasing availability in distributed storage systems via clustering," *arXiv preprint arXiv:1710.02653*, 2017.

[15] J. Sohn, B. Choi, and J. Moon, "A class of MSR codes for clustered distributed storage," *arXiv preprint arXiv:1801.02014*, 2018.

[16] I. Tamo and A. Barg, "A family of optimal locally recoverable codes," *IEEE Transactions on Information Theory*, vol. 60, no. 8, pp. 4661–4676, 2014.

[17] R. M. Roth and G. Seroussi, "On generator matrices of MDS codes (corresp.)," *IEEE transactions on information theory*, vol. 31, no. 6, pp. 826–830, 1985.