Cooperative Repair with Minimum-Storage Regenerating Codes for Distributed Storage

Jun Li, Baochun Li
Department of Electrical and Computer Engineering
University of Toronto, Canada

***Abstract*—Distributed storage systems store redundant data to tolerate failures of storage nodes and lost data should be repaired when storage nodes fail. A class of MDS codes, called *minimum-storage regenerating (MSR) codes*, has been designed to optimize bandwidth consumption when repairing one single failure. Compared with repairing failures individually, the cooperative repair of multiple failures can help to further save bandwidth consumption when multiple failures are being repaired. In this paper, we present a new construction of minimum-storage cooperative regenerating (MSCR) codes that repair two failures cooperatively and exactly. We show that given a valid instance of linear exact MSR codes, we are able to construct a corresponding repair procedure to repair any two failures cooperatively with optimal bandwidth consumption, *i.e.,* to construct an instance of exact MSCR codes directly from exact MSR codes. With this connection, we are also able to repair any single failure exactly with MSCR codes.**

I. Introduction

Distributed storage systems, *e.g.,* HDFS [1] and Windows Azure Storage [2], rely on spreading data into a large number of storage nodes that are composed from commodity hardware, to provide large-scale storage services. However, storage nodes are subject to failures, due to the large number and commodity nature of disk drives. Thus, a certain amount of redundant data should also be stored in the system. Once the data stored in some storage nodes become unavailable, they can still be fetched from the storage nodes that store the redundant data.

Conventionally, distributed storage systems use replications to produce the redundant data. For example, all data in HDFS are stored with three replicas by default. However, due to the large storage consumption of exact replicas, there is a trend for distributed storage systems to migrate from replications to erasure coding [3], [4]. Among families of erasure codes, maximum distance separable (MDS) codes achieve the optimal storage efficiency by generating coded blocks whose size is $\frac{1}{k}$ of the original data and meanwhile achieving the $[n, k]$ *recoverability property*, such that any k among a total of n coded blocks can recover the original data. For example, Facebook uses $[14, 10]$ Reed-Solomon codes [5], a classical family of MDS codes, to store non-fresh data [4], [6], saving 65% of the storage space when tolerating the same number of node failures, as compared to replication.

Once a storage node fails, it should be replaced by a new storage node, called a *newcomer*. A newcomer downloads data from existing storage nodes to repair the lost data. The amount of downloaded data, called *repair bandwidth*, is an important performance metrics of erasure codes. The repair bandwidth of some traditional MDS codes such as Reed-Solomon codes is at least the size of k blocks, *i.e.,* the size of the original data, even if only one coded block is to be repaired. However, Dimakis *et al.* [7] have shown that the repair bandwidth can be approximately the size of just one coded block, and established an optimal tradeoff between the storage, *i.e.,* the size of the coded block, and the repair bandwidth for any single-loss repair. This tradeoff can be achieved by a family of erasure codes called *regenerating codes*, while still maintaining the recoverability property.

Moreover, it is a common case to have multiple failures to repair at the same time in large-scale distributed storage systems [4]. Some distributed storage systems, such as Total Recall [8], even deliberately repair multiple failures in batches to avoid unnecessary repairs incurred by temporary node failures. If multiple newcomers can cooperate, they will enjoy a better repair bandwidth than that without cooperation [9]. The family of codes that achieve the optimal tradeoff between the storage and the repair bandwidth in cooperative repair with multiple newcomers are thus called *cooperative regenerating codes* [10], which also maintain the recoverability property. Obviously, regenerating codes defined in the single-newcomer repair can be regarded as a special case of cooperative regenerating codes with only one newcomer.

In the tradeoff between the storage and the repair bandwidth, there are two extreme points of special interest: *minimum-storage* cooperative regenerating (MSCR) codes and *minimum-bandwidth* cooperative regenerating (MBCR) codes. When there is only one newcomer during repair, the corresponding families of cooperative regenerating codes are specifically termed as MSR and MBR codes as well. Though MBCR codes achieve the minimum repair bandwidth, they sacrifice the storage efficiency since each coded block contains more than $\frac{1}{k}$ of the original data, while MSCR codes belong to the family of MDS codes by achieving both the optimal storage efficiency and the recoverability property.

Compared to just maintaining the recoverability property, it is more desirable to have exact cooperative regenerating codes such that any coded block can be repaired exactly, making it very convenient to let the corresponding codes to be systematic, so that the original data can be embedded explicitly in coded blocks. Suppose that the constructed codes support $[n, k]$ recoverability property and t newcomers download data

978-1-4799-3360-0/14/$31.00 ©2014 IEEE

from d storage nodes during repair ($n \geq d+t$, $d \geq k$). Wang *et al.* [11] have proposed an explicit construction of exact MBCR codes over all possible values of parameters $[n, k, d, t]$. However, efficiency as MDS codes, there are only a few explicit constructions of exact MSCR codes with specific values of some parameters, *e.g.,* $[n, k, d \geq 2k-2, t = 1]$ (*i.e.,* $[n, k, d \geq 2k-2]$ MSR codes) [12] and $[n, k, d = k, t \geq 1]$ [10].

On the other hand, all constructions of cooperative regenerating codes are designed for particular values of parameters, such that once coded blocks are generated they must be repaired with a specified number of storage nodes and newcomers. For example, to repair a coded block generated by $[n = 6, k = 3, d = 5]$ MSR codes, the newcomer must connect to all remaining five storage nodes. Once two coded blocks are not available, they can not be repaired with the repair bandwidth of the corresponding MSCR codes, since now there are only four storage nodes remaining, even though they are still sufficient to recover the original data. As for cooperative regenerating codes with $t > 1$, the repair procedure can occur when there are at least t node failures, and thus the storage system has no way to perform emergent repairs with fewer than t newcomers with the corresponding repair bandwidth.

In this paper, we discuss the construction of exact minimum-storage cooperative regenerating codes for all possible values of n, k, d, with one or two newcomers (*i.e.,* $t = 1$ or 2). We show that when $d - 1 \geq k$ we can build $[n, k, d-1, 2]$ exact MSCR codes directly from $[n, k, d]$ exact MSR codes, and vice versa. Based on this connection, we can either repair one failure from any d storage nodes or repair two newcomers cooperatively from any $d-1$ storage nodes. Thus even though two of coded blocks generated by $[6, 3, 5]$ MSR codes are lost, we can still repair them with the four remaining storage nodes by the repair procedure of the corresponding $[6, 3, 4, 2]$ MSCR codes, and both of these two repair procedures achieve the repair bandwidth of corresponding MSCR codes. In addition, we are able to show that there exists no linear scalar exact $[n, k, d, t = 2]$ MSCR codes if $d < 2k - 4$ and $k > 3$.

II. An Illustrative Example: Connection between MSR Codes and MSCR Codes

In this section, we give an example of the connection between MSR codes and MSCR codes. Suppose that a file contains four segments A_1, A_2, B_1 and B_2, and the data in each segment can be regarded as a vector of symbols over a finite field. In this example, we take $\mathbb{F}_8$ as the finite field, and use the exact $[4, 2, 3]$ MSR codes constructed in [13]. Thus, coded data are distributed to a total of four storage nodes. As shown in Fig. 1, each storage node stores one block containing two coded segments that are linear combinations of the four original segments, and it is easy to see that any two of them can recover the four original segments.

Since $d = 3$ in this example, a newcomer needs to download data from all the three remaining storage nodes, termed as *providers* during repair, when one storage node fails to work. Compared to traditional erasure codes that require the newcomer to download all data of providers, MSR codes only require to download a portion of data stored in each provider. In this example, the newcomer downloads only half of the data, *i.e.,* one coded segments, from each provider and then reconstructs the lost data.

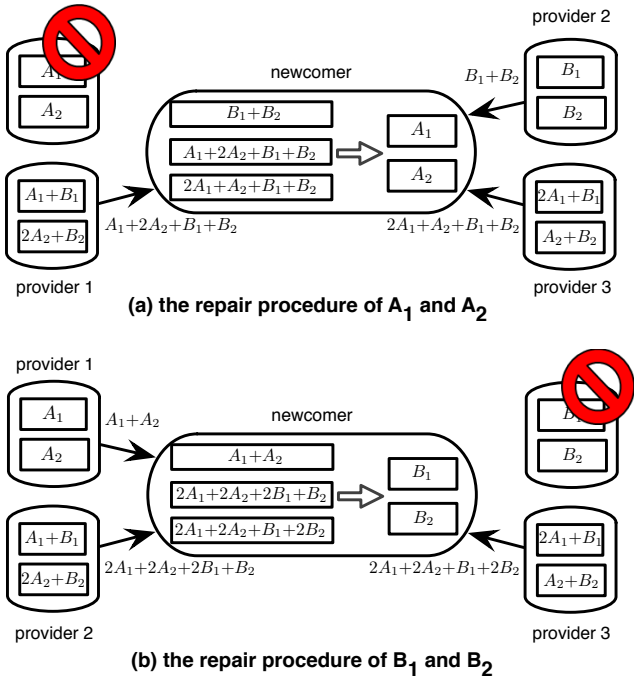

Fig. 1. An instance of exact $[4, 2, 3]$ MSR codes and two individual repair procedures of two different node failures.

Suppose that A_1 and A_2 become not accessible. As shown in Fig. 1(a), the newcomer downloads B_1+B_2, $A_1+2A_2+B_1+B_2$ and $2A_1+A_2+B_1+B_2$ from three providers, respectively. Since three downloaded segments all contain B_1+B_2, we can subtract B_1+B_2 from $A_1+2A_2+B_1+B_2$ and $2A_1+A_2+B_1+B_2$, and then reconstruct A_1 and A_2. Fig. 1(b) shows the procedure to repair B_1 and B_2, and the coded segments stored in the other two storage nodes can be repaired similarly.

When two nodes fail, we can no longer repair the lost data of each node by downloading three coded segments, since the $[4, 2, 3]$ MSR codes ask the newcomer to download data from three different providers. The only way to repair each individual newcomer is to let each newcomer download all four segments from the two remaining storage nodes, incurring additional repair bandwidth of one more segment than the repair procedures shown in Fig. 1.

In this paper, we will show that we can actually repair any two node failures in any instance of $[n, k, d]$ MSR codes cooperatively and exactly. In the cooperative repair procedure, multiple newcomers cooperate by exchanging data they have received from providers. Fig. 2 illustrates how we can cooperatively repair two newcomers, which receive data from two providers. We can see that each newcomer receives the same data just as what they receive in the individual repair. Since now there are only two providers, the two newcomers can not

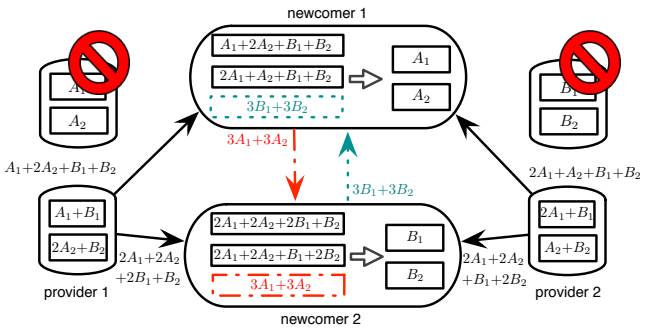

Fig. 2. The cooperative repair procedure of two node failures in the instance of exact $[4, 2, 3]$ MSR codes in Fig. 1.

recover the data of the corresponding failed node. However, the one additional segment that each newcomer needs can actually be obtained from the other newcomer. For example, in Fig. 2, the second newcomer can send $3B_1 + 3B_2$ as a linear combination of its two received symbols (as $2 + 2 = 0$ on $\mathbb{F}_8$) to the first newcomer, and thus the first newcomer can still recover A_1 and A_2 by subtracting $B_1 + B_2$ from received symbols. Similarly, the first newcomer can also send a linear combination of the two segments it has received (*i.e.,* $3A_1 + 3A_2$) to the second newcomer, such that the second newcomer can recover B_1 and B_2. We will present how to derive the segments the newcomers exchange in this paper.

In this way, we construct the exact repair procedure of each pair of two storage nodes of an instance of exact $[4, 2, 3]$ MSR codes. In other words, we have constructed an instance of exact $[4, 2, 2, 2]$ MSCR codes directly from MSR codes. As a matter of fact, this instance of $[4, 2, 2, 2]$ MSCR codes happens to coincide with the instance given in [10]. In this paper, we will show that this result can actually be generally applied to all instances of linear scalar exact MSR codes. From this connection, we are also able to repair any single node failure exactly in $[n, k, d, t = 2]$ exact MSCR code.

III. Related Work

A. Regenerating codes

Suppose that each of n storage node stores a coded block of size α symbols over a finite field $\mathbb{F}_q$ while the original file contains M symbols, and the n coded blocks achieve the $[n, k]$ recoverability property such that any k among them can recover the original data. Once one storage node fails, a newcomer downloads β symbols from each of d providers ($d \geq k$), and thus the repair bandwidth γ equals $d\beta$ symbols. The concept of regenerating codes was first proposed by Dimakis *et al.* [7], which achieves the minimum repair bandwidth for fixed n, k, d and α, while maintaining the $[n, k]$ recoverability property.

In the family of regenerating codes, the storage α and the repair bandwidth γ can not be minimized simultaneously, *i.e.,* there is a tradeoff between α and γ. The two extreme points in the tradeoff are termed as *minimum-storage* regenerating (MSR) codes and *minimum-bandwidth* regenerating (MBR) codes, respectively. The storage α and the repair bandwidth γ of MSR and MBR codes are $(\alpha_{\text{MSR}}, \gamma_{\text{MSR}}) = \left(\frac{M}{k}, \frac{Md}{k(d-k+1)}\right)$ and $(\alpha_{\text{MBR}}, \gamma_{\text{MBR}}) = \left(\frac{2Md}{k(2d-k+1)}, \frac{2Md}{k(2d-k+1)}\right)$, respectively. Obviously, the repair bandwidth of both MBR and MSR codes goes to $\frac{M}{k}$ symbols if $d \to \infty$, which is the size of the coded block in MSR codes.

The constructions of MSR and MBR codes have attracted a lot of attention. In the constructions of regenerating codes, there are two modes of repair that can be implemented in terms of the repaired data: the *functional* repair and the *exact* repair. Compared with the functional repair (*e.g.,*[14], [15]) which does not repair the lost data exactly but only preserves the recoverability property, the exact repair can exactly repair the lost data and thus it allows regenerating codes to be systematic, *i.e.,* the original data are contained explicitly in coded blocks. Systematic codes can significantly reduce the overhead of data access as no decoding operations are needed. Thus, exact regenerating codes that support the exact repair is more favored by distributed storage systems.

As for the exact repair, Rashmi *et al.* [12] have first proposed a product-matrix framework which provides an explicit construction of exact $[n, k, d]$ MBR codes for any feasible values of n, k, d ($k \leq d < n$). The construction of exact MSR codes is more complicated. By using the product-matrix framework, exact $[n, k, d]$ MSR codes can be constructed when $d \geq 2k - 2$. It has been shown that for all $[n, k, d]$, exact MSR codes can be asymptotically constructed with the symbol extension [16], such that the repair bandwidth of MSR codes can be achieved asymptotically by dividing each segment in Fig. 1 into arbitrarily small segments. Due to the symbol extension, the constructed MSR codes are not scalar and thus not practical to implement. However, it has been proved that no linear scalar exact MSR codes exist if $d < 2k - 3$ and $k > 3$ [17]. Moreover, Shah *et al.* [18] have shown that no exact regenerating codes exist that achieve interior points in the storage-bandwidth tradeoff.

B. Cooperative regenerating codes

The original storage-bandwidth tradeoff that defines the family of regenerating codes is derived under the assumption that repairs are considered individually, *i.e.,* node failures are repaired one by one in different rounds of repairs. However, Hu *et al.* [9] have first found that the repair bandwidth can have a better lower bound with multiple newcomers that cooperate with each other *w.r.t.* the storage α (for fixed n, k, d and the number of newcomers). Shum [10] and Kermarrec *et al.* [19] have independently identified the storage-bandwidth tradeoff with multiple cooperative newcomers during repair and the erasure codes achieving such repair bandwidth in this tradeoff are termed as *cooperative regenerating codes*. Supposing that there are t newcomers during repair, the storage α and the repair bandwidth γ per newcomer of minimum-storage cooperative regenerating (MSCR) codes and minimum-bandwidth cooperative regenerating (MBCR) codes

are $(\alpha_{\text{MSCR}}, \gamma_{\text{MSCR}}) = (\frac{M}{k}, \frac{M}{k} \cdot \frac{d+t-1}{d+t-k})$ and $(\alpha_{\text{MBCR}}, \gamma_{\text{MBCR}} = (\frac{M}{k} \cdot \frac{2d+t-1}{2d+t-k}, \frac{M}{k} \cdot \frac{2d+t-1}{2d+t-k})$, respectively. In this sense, regenerating codes can be regarded as a special case of cooperative regenerating codes with $t = 1$.

Wang *et al.* [11] have proposed an explicit construction of $[n, k, d, t]$ exact MBCR codes for all feasible values of $n, k, d,$ and t. However, there exist only a few constructions of exact MSCR codes for particular values of some parameters. For example, Shum *et al.* [10] proposed an construction of exact MSCR codes when $d = k$. If $k = 2$, Le Scouarnec [20] showed another construction of exact MSCR codes. Recently, Chen and Shum [21] presented an construction of $[n = 2k, k, d = 2k-2, t = 2]$ MSCR codes built from an particular construction of MSR codes. In this paper, we go a much further step by establishing a fundamental connection between exact MSR codes and exact MSCR codes for any linear scalar constructions.

IV. Definitions

Suppose that the original data has M symbols over a finite field $\mathbb{F}_q$ of size q and coded blocks are constructed over this finite field as well. We want to maintain the $[n, k]$ recoverability property, such that there are n coded blocks in total and any k of them can be used to recover the original data. Each coded block has α symbols, such that $k\alpha \geq M$.

Minimum-Storage Regenerating (MSR) codes and Minimum-Storage Cooperative Regenerating (MSCR) codes are both derived from the minimum-storage point in the storage-bandwidth tradeoff [7], by taking the storage α to the theoretical minimum, *i.e.,* $\alpha = \frac{M}{k}$.

A. Minimum-storage regenerating (MSR) codes

We define the coding schemes of exact MSR codes first. As for MSR codes, $\beta = \frac{Md}{k(d-k+1)}$, *i.e.* the newcomer needs to download β symbols from each of d providers, $d \geq k$. If we have $[n, k, d]$ MSR codes with parameters $(M, \alpha = \frac{M}{k}, \beta = \frac{Md}{k(d-k+1)})$, we can easily construct $[n, k, d]$ MSR codes with parameters $(\sigma M, \sigma\alpha, \sigma\beta)$ by dividing the σM symbols in the original data into σ groups of M symbols, where σ is a positive integer. If there exists a construction of MSR codes such that $\beta = 1$, the corresponding MSR codes is termed as *scalar* MSR codes. When $d < 2k-3$ and $k > 3$, there exists no linear scalar construction of exact MSR codes [17]. We consider the linear scalar construction only in this paper, and thus suppose that $\beta = 1$ for simplicity.

When $\beta = 1$, we can get that $M = k(d-k+1)$ and $\alpha = d-k+1$. In other words, the original data contain $k(d-k+1)$ symbols over $\mathbb{F}_q$. We use $\mathsf{M} = (m_1, \ldots, m_M)^T$ to denote the M symbols of the original data. Each coded block can be generated by a $((d-k+1) \times M)$ generating matrix Φ_i such that $c_i = \Phi_i \mathsf{M}$. Without loss of generality, we suppose that c_i is stored in the i-th storage node, $i = 1, \ldots, n$. To achieve the $[n, k]$ recoverability property, for any k-subset $S = \{s_1, \ldots, s_k\}$ of $\{1, \ldots, n\}$, $\Phi_S = \begin{pmatrix} \Phi_{s_1} \\ \vdots \\ \Phi_{s_k} \end{pmatrix}$ is invertible, *i.e.,* M can be recovered by solving the following linear system:

$$\Phi_S \cdot \mathsf{M} = \begin{pmatrix} c_{s_1} \\ \vdots \\ c_{s_k} \end{pmatrix}.$$

For example, Fig. 1 illustrates an instance of $[4, 2, 3]$ MSR codes. Suppose that the original message contains four symbols as A_1, A_2, B_1 and B_2, *i.e.,* $\mathsf{M} = \begin{pmatrix} A_1 & A_2 & B_1 & B_2 \end{pmatrix}^T$. Thus the generating matrices of the four coded blocks are $\Phi_1 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}$, $\Phi_2 = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$, $\Phi_3 = \begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 2 & 0 & 1 \end{pmatrix}$, and $\Phi_4 = \begin{pmatrix} 2 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{pmatrix}$.

Suppose that the i-th coded block c_i becomes not available, $1 \leq i \leq n$. To repair c_i, the newcomer downloads data from d providers $P = \{p_1, \ldots, p_d\}$, which is a d-subset of $\{1, \ldots, n\} \setminus \{i\}$. The provider p_j sends a linear combination of its data, *i.e.,* $\lambda_{P,i}^T(p_j) \cdot c_{p_j}$, to the newcomer where $\lambda_{P,i}(p_j)$ is a $((d-k+1) \times 1)$ vector over $\mathbb{F}_q$ and the superscript T denotes the transpose. In $\lambda_{P,i}(p_j)$, the superscript represents the procedure that repairs c_i from providers in P and the parameter p_j represents that this is a vector of linear combination coefficients of provider p_j. In this way, each provider sends one symbol over $\mathbb{F}_q$ to the newcomer.

In this paper, we define an operator $\circ_d$ to represent an operation of two matrices. Suppose that A and B are both matrices that can be equally partitioned into d row groups, *i.e.,*

$$A = \begin{pmatrix} A_1 \\ \vdots \\ A_d \end{pmatrix}, \text{ and } B = \begin{pmatrix} B_1 \\ \vdots \\ B_d \end{pmatrix}.$$

Then we define $A \circ_d B$ as

$$A \circ_d B = \begin{pmatrix} A_1 B_1 \\ \vdots \\ A_d B_d \end{pmatrix}.$$

In this sense, the operator $\circ_d$ can be regarded as a blockwise extension of the Hadamard product. Specifically, $A \circ_d B = A \cdot B$ when $d = 1$.

Let $\Lambda_{P,i} = \begin{pmatrix} \lambda_{P,i}^T(p_1) \\ \vdots \\ \lambda_{P,i}^T(p_d) \end{pmatrix}$ and $\Phi_P = \begin{pmatrix} \Phi_{p_1} \\ \vdots \\ \Phi_{p_d} \end{pmatrix}$. The data that the newcomer received from providers can be represented as $(\Lambda_{P,i} \circ_d \Phi_P)\mathsf{M}$. The newcomer needs to encode the received data with a $d \times (d-k+1)$ matrix $\delta_{P,i}$ to repair c_i exactly, such that $c_i = \delta_{P,i}^T(\Lambda_{P,i} \circ_d \Phi_P)\mathsf{M}$, *i.e.,*

$$\Phi_i = \delta_{P,i}^T(\Lambda_{P,i} \circ_d \Phi_P). \quad (1)$$

In this sense, we also use Φ_i in this paper to represent c_i during repair.

Given a linear scalar instance of exact $[n, k, d]$ MSR codes with generating matrices Φ_i, $i = 1, \ldots, n$, there exists

$(\Lambda_{P,i}, \delta_{P,i})$ such that (1) holds, $\forall i \in \{1,\ldots,n\}$ and $\forall$ d-subset $P \subset \{1,\ldots,n\} \setminus \{i\}$.

Fig. 1(a) shows the procedure to repair Φ_1 of the corresponding instance of $[4,2,3]$ exact MSR codes, *i.e.,* $i = 1$ and $P_2 = \{2,3,4\}$. Then $\Lambda_{P_2,1} = \begin{pmatrix} 1 & 1 \\ 1 & 1 \\ 1 & 1 \end{pmatrix}$ and $\delta_{P_2,1} = \begin{pmatrix} 6 & 6 \\ 2 & 4 \\ 4 & 2 \end{pmatrix}$. Then we can validate that (1) holds for this example. Notice again that all linear operations are over $\mathbb{F}_8$ and the primitive polynomial we use in this example is x^3+x+1.

B. Minimum-storage cooperative regenerating (MSCR) codes

Assume that we have d providers and t newcomers during repair where $d \geq k$, and $t \geq 1$. In the cooperative repair procedure, the newcomer first downloads β_1 symbols from each provider and then receives β_2 symbols from each of other $t-1$ newcomers. As for MSCR codes, $\beta_1 = \beta_2 = \frac{M}{k(d-k+t)}$. Therefore, the repair bandwidth of MSCR codes per newcomer is $\frac{M(d-1+t)}{k(d-k+t)}$ symbols. Once again, we can regard MSCR codes as an extension of MSR codes for $t \geq 1$. When $t = 1$, the newcomer needs to receive data from providers only, as there is no other newcomer, and the repair bandwidth for this newcomer is exactly the repair bandwidth of MSR codes.

Like MSR codes, we discuss the linear scalar construction of $[n,k,d,t]$ exact MSCR codes. Thus, we still let $\beta_1 = \beta_2 = 1$ for simplicity, and then we get that $M = k(d-k+t)$ and $\alpha = d-k+t$. Similar to the coding scheme of MSR codes, we let $\mathsf{M} = (m_1,\ldots,m_M)^T$ to represent the original data. Each coded block is generated by a $(d-k+t)\times M$ generating matrix Φ_i such that $c_i = \Phi_i\mathsf{M}$, $i = 1,\ldots,n$. For any k-subset S of $\{1,\ldots,n\}$, Φ_S is invertible, such that the $[n,k]$ recoverability property is achieved. Obviously, an instance of $[n,k,d']$ MSR codes can coincide with an instance of $[n,k,d,t]$ MSCR codes, if $d' = d+t$, as shown in Fig. 1 and Fig. 2.

We suppose that $c_{l_1},\ldots,c_{l_t}$ are not available, where $L = \{l_1,\ldots,l_t\}$ is a t-subset of $\{1,\ldots,n\}$. Without loss of generality, we let the newcomer l_i repair c_{l_i}, $i = 1,\ldots,t$. Besides, there are d providers $p_1,\ldots,p_d$ selected from $n-t$ existing nodes, *i.e.,* $P \subset \{1,\ldots,n\} \setminus L$ where $P = \{p_1,\ldots,p_d\}$. The provider p_j sends $\lambda^T_{P,L}(p_j,l_i)\cdot c_{p_j}$ to the newcomer l_i, $i = 1,\ldots,t$, $j = 1,\ldots,d$. $\lambda_{P,L}(p_j,l_i)$ is a $((d-k+t)\times 1)$ vector over $\mathbb{F}_q$, where the superscript denotes the procedure to repair newcomers in L with providers in P and its parameters mean that this is the vector of linear combination coefficients of the symbol that p_j sends to l_i. Therefore, the newcomer l_i can get d symbols from all providers, *i.e.,* $\lambda^T_{P,L}(p_j,l_i)\cdot c_{p_j}$, $j = 1,\ldots,d$.

Then the newcomer l_i sends a linear combination of received symbols to each of other newcomers with coefficients $\pi_{P,L}(l_i,l_h)$, $1 \leq i \leq t$, $1 \leq h \leq t$, $i \neq h$. Let $\Lambda_{P,L}(l_i) = \begin{pmatrix} \lambda^T_{P,L}(p_1,l_i) \\ \vdots \\ \lambda^T_{P,L}(p_d,l_i) \end{pmatrix}$, which contains all coefficients of symbols that the newcomer l_i receives from providers. Since $\lambda^T_{P,L}(p_j,l_i)\cdot c_{p_j} = \lambda^T_{P,L}(p_j,l_i)\cdot\Phi_{p_j}\mathsf{M}$, the newcomer l_i sends $\pi^T_{P,L}(l_i,l_h)(\Lambda_{P,L}(l_i)\circ_d\Phi_P)\mathsf{M}$ to the newcomer l_h, where $\pi_{P,L}(l_i,l_h)$ is a $d\times 1$ vector on $\mathbb{F}_q$, $h \in \{1,\ldots,t\}\setminus\{i\}$. In return, the newcomer l_i also receives symbols from other newcomers, *i.e.,* $\pi^T_{P,L}(l_h,l_i)(\Lambda_{P,L}(l_h)\circ_d\Phi_P)\mathsf{M}$, $h \in \{1,\ldots,t\}\setminus\{i\}$.

Let $\Pi_{P,L}(l_i) = \begin{pmatrix} \pi^T_{P,L}(l_1,l_i) \\ \vdots \\ \pi^T_{P,L}(l_{i-1},l_i) \\ \pi^T_{P,L}(l_{i+1},l_i) \\ \vdots \\ \pi^T_{P,L}(l_t,l_i) \end{pmatrix}$, and $\Psi_{P,L}(l_i) = \begin{pmatrix} \Lambda_{P,L}(l_1)\circ_d\Phi_P \\ \vdots \\ \Lambda_{P,L}(l_{i-1})\circ_d\Phi_P \\ \Lambda_{P,L}(l_{i+1})\circ_d\Phi_P \\ \vdots \\ \Lambda_{P,L}(l_t)\circ_d\Phi_P \end{pmatrix}$. With the d symbols received from providers and $t-1$ symbols received from other newcomers, the newcomer l_i can repair c_i exactly by computing $\delta^T_{P,L}(l_i)\begin{pmatrix} (\Lambda_{P,L}(l_i)\circ_d\Phi_P)\mathsf{M} \\ (\Pi_{P,L}(l_i)\circ_{t-1}\Psi_{P,L}(l_i))\mathsf{M} \end{pmatrix}$, where $\delta_{P,L}(l_i)$ are $(d-1+t)\times(d-k+t)$ matrix on $\mathbb{F}_q$. Hence, $c_{l_i} = \delta^T_{P,L}(l_i)\begin{pmatrix} (\Lambda_{P,L}(l_i)\circ_d\Phi_P)\mathsf{M} \\ (\Pi_{P,L}(l_i)\circ_{t-1}\Psi_{P,L}(l_i))\mathsf{M} \end{pmatrix}$, *i.e.,*

$$\Phi_{l_i} = \delta^T_{P_0,L}(l_i)\begin{pmatrix} \Lambda_{P_0,L}(l_i)\circ_d\Phi_P \\ \Pi_{P_0,L}(l_i)\circ_{t-1}\Psi_{P,L}(l_i) \end{pmatrix}. \quad (2)$$

Given a linear scalar instance of $[n,k,d,t]$ exact MSCR codes with generating matrices Φ_i, $i = 1,\ldots,n$, there exists $(\Lambda_{P,L}(l_i),\Pi_{P,L}(l_i),\delta_{P,L}(l_i))$ for all $i \in \{1,\ldots,t\}$, $\forall$ t-subset $L \subset \{1,\ldots,n\}$, $\forall$ d-subset $P \subset \{1,\ldots,n\}\setminus L$, such that (2) holds.

From the instance of $[4,2,2,2]$ exact MSCR codes shown in Fig. 1(b), we can get a repair procedure with $L = \{1,2\}$ and $P_0 = \{3,4\}$, such that

$$\Lambda_{P_0,L}(1) = \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix},\ \Lambda_{P_0,L}(2) = \begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix},$$
$$\Pi_{P_0,L}(1) = \begin{pmatrix} 6 & 6 \end{pmatrix},\ \Pi_{P_0,L}(2) = \begin{pmatrix} 6 & 6 \end{pmatrix},$$
$$\delta_{P_0,L}(1) = \begin{pmatrix} 2 & 4 \\ 4 & 2 \\ 6 & 6 \end{pmatrix},\ \text{and}\ \delta_{P_0,L}(2) = \begin{pmatrix} 4 & 2 \\ 2 & 4 \\ 7 & 7 \end{pmatrix}.$$

We will show that the corresponding $(\Lambda_{P_0,L}(i),\Pi_{P_0,L}(i),\delta_{P_0,L}(i))$, $i = 1,2$, can be derived from the instance of $[4,2,3]$ MSR codes in Fig. 1(a), and vice versa.

Our definitions of exact MSR codes and exact MSCR codes can cover all scalar constructions so far, as they are all linear codes. Even though some constructions, such as the Product-Matrix construction [12], has different representations, it is easy to see that they can be transferred equivalently into our definitions, as long as the coded blocks are linear combinations of the original data over the finite field.

V. Constructing MSCR codes from MSR codes

In this paper, we will show the connection between exact MSR codes and exact MSCR codes. Our intuition comes from Fig. 1 and Fig. 2, where two instances of MSR codes and MSCR codes have the same coding instance but different repair procedures. Hence, we suppose that from an instance and its repair procedure of MSR codes or MSCR codes, we can derive the repair procedure of the other family of codes with the same instance. In this section, we discuss one side of this connection, constructing exact MSCR codes from an instance of exact MSR codes.

Given an instance of $[n, k, d]$ exact MSR codes ($n > d \geq k$), *i.e.,* $\Phi_i, i = 1, \ldots, n$, the repair procedure of any coded block $c_i = \Phi_i \mathsf{M}$ can be determined by $(\Lambda_{P,i}, \delta_{P,i})$ such that (1) holds, for any d-subset $P \subset \{1, \ldots, n\} \backslash \{i\}$. Now suppose that we have an instance of $[n, k, d-1, t=2]$ MSCR codes with the same generating matrices $\Phi_i, i = 1, \ldots, n$. Since there must be at least k providers in a cooperative repair procedure, d must be no less than $k+1$.

Without loss of generality, we assume that the set of newcomers contains two nodes, *i.e.,* $L = \{l_1, l_2\}$, $1 \leq l_1 < l_2 \leq n$. To repair Φ_{l_1} and Φ_{l_2} exactly with $d-1$ providers in $P_0 = \{p_1, \ldots, p_{d-1}\}$, we need to derive $(\Lambda_{P_0,L}(l_i), \Pi_{P_0,L}(l_i), \delta_{P_0,L}(l_i)), i = 1, 2$, such that (2) holds.

Let $P_2 = P_0 \cup \{l_2\}$. The corresponding instance of MSR codes can repair Φ_{l_1} exactly with providers in P_2, *i.e.,*

$$\begin{aligned}\Phi_{l_1} &= \delta^T_{P_2,l_1}(\Lambda_{P_2,l_1} \circ_d \Phi_{P_2}) \\ &= \begin{pmatrix} \delta_{P_2,l_1}(P_0) \\ \delta_{P_2,l_1}(l_2) \end{pmatrix}^T \cdot \left(\begin{pmatrix} \Lambda_{P_2,l_1}(P_0) \\ \lambda^T_{P_2,l_1}(l_2) \end{pmatrix} \circ_d \begin{pmatrix} \Phi_{P_0} \\ \Phi_{l_2} \end{pmatrix} \right),\end{aligned}$$

where $\delta_{P_2,l_1}(P_0)$ and $\Lambda_{P_2,l_1}(P_0)$ denotes the rows in δ_{P_2,l_1} and Λ_{P_2,l_1} that correspond to symbols received from P_0, respectively.

Since $A \circ_1 B = A \cdot B$, we can get

$$\begin{aligned}\Phi_{l_1} = \delta^T_{P_2,l_1}(P_0) \cdot (\Lambda_{P_2,l_1}(P_0) \circ_{d-1} \Phi_{P_0}) \\ + (\lambda_{P_2,l_1}(l_2)\delta_{P_2,l_1}(l_2))^T \Phi_{l_2}. \quad (3)\end{aligned}$$

Similarly, we can repair Φ_{l_2} exactly from providers in $P_1 = P_0 \cup \{l_1\}$ and have

$$\begin{aligned}\Phi_{l_2} = \delta^T_{P_1,l_2}(P_0) \cdot (\Lambda_{P_1,l_2}(P_0) \circ_{d-1} \Phi_{P_0}) \\ + (\lambda_{P_1,l_2}(l_1)\delta_{P_1,l_2}(l_1))^T \Phi_{l_1}. \quad (4)\end{aligned}$$

Combining (3) and (4), we can get

$$\begin{aligned}\Phi_{l_1} = \Delta_{P_0,L}(l_1)^{-1} \cdot \begin{pmatrix} \delta_{P_2,l_1}(P_0) \\ \delta_{P_2,l_1}(l_2) \end{pmatrix}^T \cdot \\ \begin{pmatrix} \Lambda_{P_2,l_1}(P_0) \circ_{d-1} \Phi_{P_0} \\ (\delta_{P_1,l_2}(P_0)\lambda_{P_2,l_1}(l_2))^T \circ_1 (\Lambda_{P_1,l_2}(P_0) \circ_{d-1} \Phi_{P_0}) \end{pmatrix}, \quad (5)\end{aligned}$$

where $\Delta_{P_0,L}(l_1) = I_{d-k+1} - (\lambda_{P_1,l_2}(l_1)\delta_{P_1,l_2}(l_1)\lambda_{P_2,l_1}(l_2)\delta_{P_2,l_1}(l_2))^T$ and I_{d-k+1} denotes a $(d-k+1) \times (d-k+1)$ identity matrix. Therefore, as long as $\Delta_{P_0,L}(l_1)$ is invertible, we can have a procedure with providers in P_0 to repairs both Φ_{l_1} and Φ_{l_2} exactly. In Appendix, we prove that for any instance of $[n, k, d]$ exact MSR codes where $d \geq k+1$, there exists a repair procedure such that $\Delta_{P_0,L}(l_1)$ is invertible.

By comparing (5) with (2), we can get

$$\delta^T_{P_0,L}(l_1) = \Delta^{-1}_{P_0,L}(l_1) \cdot \begin{pmatrix} \delta_{P_2,l_1}(P_0) \\ \delta_{P_2,l_1}(l_2) \end{pmatrix}^T; \quad (6)$$

$$\Lambda_{P_0,L}(l_1) = \Lambda_{P_2,l_1}(P_0); \text{ and} \quad (7)$$

$$\Pi_{P_0,L}(l_1) = (\pi^T_{P_0,L}(l_2, l_1)) = (\delta_{P_1,l_2}(P_0)\lambda_{P_2,l_1}(l_2))^T. \quad (8)$$

Moreover, since $\Psi_{P_0,L}(l_1) = \Lambda_{P_0,L}(l_2) \circ_{d-1} \Phi_{P_0}$ when $t = 2$, we can get that

$$\Lambda_{P_0,L}(l_2) = \Lambda_{P_1,l_2}(P_0). \quad (9)$$

Now we have got the linear coefficients to exactly repair Φ_{l_1}. Similarly, we can get the coefficients to repair Φ_{l_2},

$$\delta^T_{P_0,L}(l_2) = \Delta^{-1}_{P_0,L}(l_2) \cdot \begin{pmatrix} \delta_{P_1,l_2}(P_0) \\ \delta_{P_1,l_2}(l_1) \end{pmatrix}^T; \quad (10)$$

$$\Lambda_{P_0,L}(l_2) = \Lambda_{P_1,l_2}(P_0); \text{ and} \quad (11)$$

$$\Pi_{P_0,L}(l_2) = (\pi_{P_0,L}(l_1, l_2))^T = (\delta_{P_2,l_1}(P_0)\lambda_{P_1,l_2}(l_1))^T, \quad (12)$$

where

$$\Delta_{P_0,L}(l_2) = I_{d-k+1} - (\lambda_{P_2,l_1}(l_2)\delta_{P_2,l_1}(l_2)\lambda_{P_1,l_2}(l_1)\delta_{P_1,l_2}(l_1))^T.$$

Supposing that A and B are square matrices, $I - AB$ is invertible if and only if $I - BA$ is invertible [22]. Therefore, $\Delta_{P_0,L}(l_2)$ is invertible as long as if $\Delta_{P_0,L}(l_1)$ is invertible.

In addition, from $\Psi_{P_0,L}(l_2)$ we know that

$$\Lambda_{P_0,L}(l_1) = \Lambda_{P_2,l_1}(P_0). \quad (13)$$

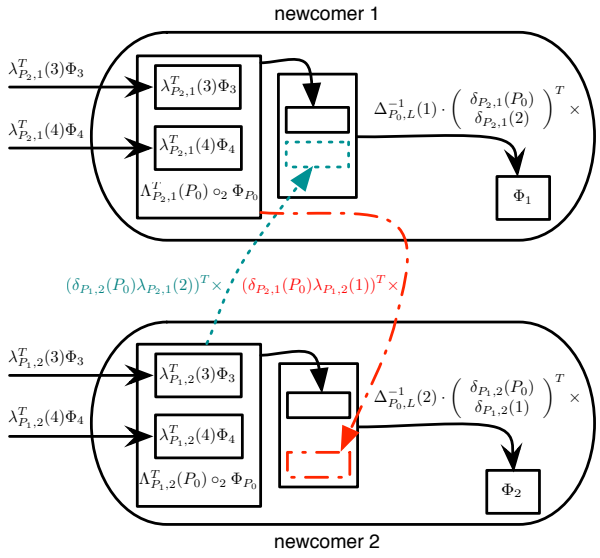

Fig. 3. The cooperative repair procedure of Φ_1 and Φ_2, derived from the exact $[4, 2, 3]$ MSR codes shown in Fig. 1.

It is interesting to see that (7) and (9) are equivalent to (13) and (11), respectively. In other words, in the two derived repair procedures the two newcomers l_1 and l_2 receive the same data from each provider and receive one symbol from each other. Therefore, we can combine them to have a cooperative repair procedure that repairs both Φ_{l_1} and Φ_{l_2} exactly.

Given the instance of $[4, 2, 3]$ exact MSR codes shown in Fig. 1, we can directly get all coded blocks as Φ_i, $i = 1, \ldots, n$, of $[4, 2, 2, 2]$ exact MSCR codes, and derive the cooperative repair procedure of any two coded blocks. For example, to repair Φ_1 and Φ_2, *i.e.,* $L = \{1, 2\}$, from providers in $P_0 = \{3, 4\}$, we can derive the corresponding repair procedure from the repair procedures of Φ_1 (with providers in $P_2 = P_0 \cup \{2\}$) and Φ_2 (with providers in $P_1 = P_0 \cup \{1\}$), as shown in Fig. 3. Notice that in Fig. 3,

$$\Lambda^T_{P_2,1}(P_0) = \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix},$$
$$\Lambda^T_{P_1,2}(P_0) = \begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix},$$
$$(\delta_{P_1,2}(P_0)\lambda_{P_2,1}(2))^T = (\delta_{P_2,1}(P_0)\lambda_{P_1,2}(1))^T = \begin{pmatrix} 6 & 6 \end{pmatrix},$$
$$\Delta_{P_0,L}(1)^{-1} \cdot \begin{pmatrix} \delta_{P_2,1}(P_0) \\ \delta_{P_2,1}(2) \end{pmatrix}^T = \begin{pmatrix} 2 & 4 \\ 4 & 2 \\ 6 & 6 \end{pmatrix}, \text{ and}$$
$$\Delta^{-1}_{P_0,L}(2) \cdot \begin{pmatrix} \delta_{P_1,2}(P_0) \\ \delta_{P_1,2}(1) \end{pmatrix}^T = \begin{pmatrix} 4 & 2 \\ 2 & 4 \\ 7 & 7 \end{pmatrix}.$$

The equations above formally describe a cooperative repair procedure that exactly corresponds to the repair procedure shown in Fig. 2.

Thus, by (6)-(9), (10) and (12), we have already get the repair procedure of $[n, k, d-1, 2]$ exact MSCR codes from an instance of $[n, k, d]$ exact MSR codes, $d \geq k+1$. In other words, we have found $(\Lambda^{P_0,L}_{l_i}, \Pi^{P_0,L}_{l_i}, \delta^{P_0,L}_{l_i})$ that makes (2) hold when i equals 1 and 2, respectively.

VI. Constructing MSR codes from MSCR codes

In this section, we discuss the other side of the connection between exact MSR codes and exact MSCR codes. Suppose that we have a linear scalar instance of $[n, k, d-1, t=2]$ exact MSCR codes where $d \geq k+1$, and then we show that we can instantly have an instance of $[n, k, d]$ exact MSR codes and corresponding repair procedures.

In a linear scalar instance of $[n, k, d-1, t=2]$ exact MSCR codes, any two nodes in $L = \{l_1, l_2\}$ can be repaired exactly from providers in $P_0 = \{p_1, \ldots, p_{d-1}\}$, $1 \leq l_1 \leq n$, $1 \leq l_2 \leq n$, $l_1 \neq l_2$, $P_0 \subset \{1, \ldots, n\} \setminus \{l_1, l_2\}$. Since $t = 2$, we can rewrite $\delta^T_{P_0,L}(l_1)$ as $\begin{pmatrix} \delta_{P_0,L}(P_0, l_1) \\ \delta_{P_0,L}(l_2, l_1) \end{pmatrix}^T$, where $\delta_{P_0,L}(P_0, l_1)$ and $\delta_{P_0,L}(l_2, l_1)$ denotes the first $d-1$ rows and the last row in $\delta_{P_0,L}(l_1)$ that correspond to the symbols received from providers and the other newcomer, respectively. By (2) we know that

$$\Phi_{l_1} = \delta^T_{P_0,L}(P_0, l_1) \cdot (\Lambda_{P_0,L}(l_1) \circ_{d-1} \Phi_{P_0}) + \delta^T_{P_0,L}(l_2, l_1)\pi^T_{P_0,L}(l_2, l_1) \cdot (\Lambda_{P_0,L}(l_2) \circ_{d-1} \Phi_{P_0}).$$

When $k = 1$, MSR and MSCR codes will become equivalent to replications. Thus, we can suppose that $k \geq 2$ for all instances of MSR codes and MSCR codes. Since $\delta^T_{P_0,L}(P_0, l_1)$ is a $(d-k+1) \times (d-1)$ matrix and has a rank of $d-k+1$, it has a right inverse $\text{rinv}(\delta^T_{P_0,L}(P_0, l_1))$, such that $\delta^T_{P_0,L}(P_0, l_1) \cdot \text{rinv}(\delta^T_{P_0,L}(P_0, l_1)) = I_{d-k+1}$. Thus, we have

$$\Lambda_{P_0,L}(l_1) \circ_{d-1} \Phi_{P_0} = \text{rinv}(\delta^T_{P_0,L}(P_0, l_1)) \cdot (\Phi_{l_1} - \delta^T_{P_0,L}(l_2, l_1)\pi^T_{P_0,L}(l_2, l_1) \cdot (\Lambda_{P_0,L}(l_2) \circ_{d-1} \Phi_{P_0})). \quad (14)$$

Similarly, to repair Φ_{l_2} we have

$$\Phi_{l_2} = \delta^T_{P_0,L}(P_0, l_2) \cdot (\Lambda_{P_0,L}(l_2) \circ_{d-1} \Phi_{P_0}) + \delta^T_{P_0,L}(l_1, l_2)\pi^T_{P_0,L}(l_1, l_2) \cdot (\Lambda_{P_0,L}(l_1) \circ_{d-1} \Phi_{P_0}). \quad (15)$$

Replace $\Lambda_{P_0,L}(l_1) \circ_{d-1} \Phi_{P_0}$ in (15) with (14), we can get a repair procedure of Φ_{l_2} from providers in $P_1 = P_0 \cup \{l_1\}$:

$$\Phi_{l_2} = \begin{pmatrix} \Delta_{P_1,l_2} \\ \delta_{P_0,L}(l_1, l_2) \end{pmatrix}^T \cdot \begin{pmatrix} \Lambda_{P_0,L}(l_2) \circ_{d-1} \Phi_{P_0} \\ \pi^T_{P_0,L}(l_1, l_2)\text{rinv}(\delta^T_{P_0,L}(P_0, l_1)) \cdot \Phi_{l_1} \end{pmatrix}, \quad (16)$$

where $\Delta^T_{P_1,l_2} = \delta^T_{P_0,L}(P_0, l_2) - \delta^T_{P_0,L}(l_1, l_2)\pi^T_{P_0,L}(l_1, l_2) \cdot \text{rinv}(\delta^T_{P_0,L}(P_0, l_1))\delta^T_{P_0,L}(l_2, l_1)\pi^T_{P_0,L}(l_2, l_1)$.

Comparing (16) with (1), we can get the exact repair procedure of Φ_{l_2} with providers in P_1, where

$$\delta_{P_1,l_2} = \begin{pmatrix} \Delta_{P_1,l_2} \\ \delta_{P_0,L}(l_1, l_2) \end{pmatrix}^T; \text{ and} \quad (17)$$
$$\Lambda_{P_1,l_2} = \begin{pmatrix} \Lambda_{P_0,L}(l_2) \\ \pi^T_{P_0,L}(l_1, l_2)\text{rinv}(\delta^T_{P_0,L}(P_0, l_1)) \end{pmatrix}. \quad (18)$$

Consider the instance of $[4, 2, 2, 2]$ exact MSCR codes which is shown in Fig. 2 and is also derived in Sec. VI, we can apply (17) and (18) to repair any single failure. For example, to repair Φ_2, *i.e.,* $l_2 = 2$, Fig. 4 illustrates the derived repair procedure where

$$\Lambda_{P_0,L}(l_2) = \begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix},$$
$$\pi^T_{P_0,L}(l_1, l_2)\text{rinv}(\delta^T_{P_0,L}(P_0, l_1)) = \begin{pmatrix} 1 & 1 \end{pmatrix}, \text{ and}$$
$$\begin{pmatrix} \Delta_{P_1,l_2} \\ \delta_{P_0,L}(l_1, l_2) \end{pmatrix} = \begin{pmatrix} 4 & 2 \\ 2 & 4 \\ 7 & 7 \end{pmatrix}.$$

The repair procedure described above corresponds to the repair procedure shown in Fig. 1(b).

Therefore, $\forall\ l_1 \neq l_2$, we can derive the repair procedure of Φ_{l_2} from the cooperative repair procedure of Φ_{l_1} and Φ_{l_2}. In this way, we can repair any single failure exactly in

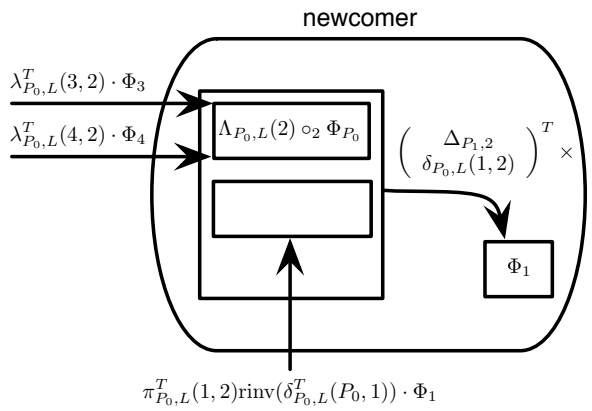

Fig. 4. The repair procedure of Φ_2, derived from the instance of $[4,2,2,2]$ MSCR codes in Fig. 2.

$[n,k,d,t=2]$ MSCR codes. In other words, from an instance of $[n,k,d,t=2]$ MSCR codes we can get an instance of $[n,k,d+1]$ exact MSR codes directly and derive the repair procedure of any Φ_i, $i=1,\ldots,n$.

VII. Discussions

We have established the connection between exact MSR codes and exact MSCR codes in Sec. V and Sec. VI. In this section, we discuss some interesting properties of MSCR codes derived from this connection.

A. Construction of $[n,k,d,t=2]$ MSCR codes

In this paper, we show that given an instance of $[n,k,d]$ exact MSR codes ($d\geq k+1$), we can instantly construct an instance of $[n,k,d-1,t=2]$ exact MSCR codes. This means that we have broadly expanded the parameters of exact MSCR codes that have explicit constructions. To our best knowledge, there exists a construction of exact MSR codes as long as $d\geq 2k-2$ [12]. Therefore, if $n\geq d+1$ and $d\geq k$, we can get $[n,k,d,t=2]$ exact MSCR codes as long as $d\geq 2k-3$.

On the other hand, we also show that given a scalar construction of linear exact MSCR codes, we can derive a construction of exact MSCR codes. Since we know that there exists no scalar construction of $[n,k,d]$ linear exact MSR codes when $d<2k-3$ and $k>3$ [17], it is impossible to have scalar construction of $[n,k,d,t=2]$ exact MSCR codes when $d<2k-4$ and $k>3$.

To summarize, we discuss the construction of $[n,k,d,t=2]$ exact MSCR codes for all possible values of $[n,k]$. We show the existence or the non-existence of linear scalar constructions of all possible values of d, except the only open case of $d=2k-3$, where $k>3$.

B. Flexible repair of MSCR codes

Based on the connection between exact MSR codes and exact MSCR codes, when $d\geq k+1$, the set of linear scalar instances of $[n,k,d]$ exact MSR codes and $[n,k,d-1,t=2]$ exact MSCR codes are equivalent. In other words, given a linear scalar instance of exact MSR codes, there is the same instance of exact MSCR codes with the corresponding parameters, and vice versa. When $d=k$, on the other hand, there exists no instance of $[n,k,d-1,t=2]$ MSCR codes, since there must be at least k providers during repair.

In fact, the equivalence between exact MSR codes and exact MSCR codes makes it possible to construct exact MSCR codes directly from linear scalar MSR codes, by repairing any two failures cooperatively. In this sense, we can achieve a much more flexible repair procedure than the repair procedure of all constructions of regenerating codes proposed previously. Conventionally, once an instance of MSR codes or MSCR codes has been constructed, it is supposed to repair a fixed number of t newcomers ($t=1$ for MSR codes). In this paper, we show that we can repair not only one failure, but two failures in exact MSR codes as well. In other words, due to the equivalence between MSR codes and MSCR codes, we can also repair any single failure in MSR codes. This property makes it possible to achieve a flexible repair procedure with either one or two newcomers in distributed storage systems.

C. Inheriting advantages of MSR codes

Because the MSCR codes constructed in this paper are derived from MSR codes, we can directly inherit advantages of the corresponding instance of MSR codes. The most straightforward and important advantage inherited from MSR codes is that the derived instance of MSCR codes can be systematic, *i.e.,* the original data are embedded into coded blocks. Systematic MSCR codes can help to significantly reduce the access latency and the exact repair makes the repaired data remain systematic after any rounds of repair procedures.

Some particular constructions of MSR codes can have extra advantages. For example, Han *et al.* [23] have presented a construction of MSR codes with the optimal update complexity. Since we can build cooperative repair procedures from this instance of MSR codes, the derived MSCR codes can also enjoy this advantage of efficient updating.

VIII. Conclusion

In this paper, we present a connection between exact MSR codes and exact MSCR codes, such that we can build exact MSCR codes directly from any instance of linear scalar exact MSR codes. From this connection, we know that when $d\geq k+1$, the set of instances of linear scalar $[n,k,d-1,t=2]$ exact MSCR codes and linear scalar $[n,k,d]$ exact MSR codes are equivalent. In other words, we propose an explicit construction of $[n,k,d,t=2]$ MSCR codes for $d\geq 2k-3$ and show that there exists no explicit construction when $d<2k-4$ and $k>3$. Since the constructed MSCR codes are directly derived from corresponding MSR codes, we can achieve a flexible repair procedure that repairs any one or two failures in MSR codes as well.

APPENDIX
PROOF OF THAT $\Delta_{P_0,L}(l_1)$ IS INVERTIBLE.

As

$$\begin{aligned}\Delta_{P_0,L}(l_1) &= I_{d-k+1} - (\lambda_{P_1,l_2}(l_1)\delta_{P_1,l_2}(l_1)\lambda_{P_2,l_1}(l_2)\delta_{P_2,l_1}(l_2))^T, \\ &= I_{d-k+1} - \delta^T_{P_2,l_1}(l_2)\lambda^T_{P_2,l_1}(l_2)\delta^T_{P_1,l_2}(l_1)\lambda^T_{P_1,l_2}(l_1),\end{aligned} \tag{19}$$

$\Delta_{P_0,L}(l_1)$ is non-invertible if and only if 1 is an eigenvalue of $\delta^T_{P_2,l_1}(l_2)\lambda^T_{P_2,l_1}(l_2)\delta^T_{P_1,l_2}(l_1)\lambda^T_{P_1,l_2}(l_1)$.

Since the rank of $\delta^T_{P_2,l_1}(l_2)\lambda^T_{P_2,l_1}(l_2)\delta^T_{P_1,l_2}(l_1)\lambda^T_{P_1,l_2}(l_1)$ is 1, the only non-zero eigenvalue is $\lambda^T_{P_2,l_1}(l_2)\delta^T_{P_1,l_2}(l_1)\lambda^T_{P_1,l_2}(l_1)\delta^T_{P_2,l_1}(l_2)$. We prove that we can have an arbitrary $\delta_{P_2,l_1}(l_2)$ such that the non-zero eigenvalue is not 1.

By (1), we have

$$\Phi_{l_1} = \sum_{j=0}^{d-1} \delta^T_{P_2,l_1}(p_j)\lambda^T_{P_2,l_1}(p_j)\Phi_{p_j},$$

where $\delta_{P_2,l_1}(p_j)$ denotes the row in δ_{P_2,l_1} that corresponds to the symbol received from provider p_j. Without loss of generality, let $p_0 = l_2$ and thus $\delta^T_{P_2,l_1}(p_0) = \delta^T_{P_2,l_1}(l_2)$.

Given $\lambda^T_{P_2,l_1}(l_2)\delta^T_{P_1,l_2}(l_1)\lambda^T_{P_1,l_2}(l_1)$, we can find $\tilde{\delta}^T_{P_2,l_1}(p_0)$ such that $\lambda^T_{P_2,l_1}(l_2)\delta^T_{P_1,l_2}(l_1)\lambda^T_{P_1,l_2}(l_1)\tilde{\delta}^T_{P_2,l_1}(l_2) \neq 1$. When $d \geq k+1$, there exist $\tilde{\delta}^T_{P_2,l_1}(p_j)$, $j = 1, \ldots, d-1$, such that

$$\begin{aligned}&(\delta^T_{P_2,l_1}(p_0) - \tilde{\delta}^T_{P_2,l_1}(p_0))\lambda^T_{P_2,l_1}(p_0)\Phi_{p_0} = \\ &\qquad \sum_{j=1}^{d-1}(\delta^T_{P_2,l_1}(p_j) - \tilde{\delta}^T_{P_2,l_1}(p_j))\lambda^T_{P_2,l_1}(p_j)\Phi_{p_j},\end{aligned} \tag{20}$$

because $\begin{pmatrix} \Phi_{p_1} \\ \vdots \\ \Phi_{p_{d-1}} \end{pmatrix}$ has full-rank, which is guaranteed by the recoverability property of MSR codes.

By (20), we know that

$$\Phi_{l_1} = \sum_{j=0}^{d-1} \tilde{\delta}^T_{P_2,l_1}(p_j)\lambda^T_{P_2,l_1}(p_j)\Phi_{p_j}.$$

Now we have another repair procedure of Φ_{l_1} and then we can replace $\delta^T_{P_2,l_1}(p_j)$ with $\tilde{\delta}^T_{P_2,l_1}(p_j)$ in (19) such that $\Delta_{P_0,L}(l_1)$ is invertible.

Therefore, there exists a repair procedure such that $\Delta_{P_0,L}(l_1)$ is invertible, for any linear scalar instance of $[n,k,d]$ MSR codes when $d \geq k+1$.

REFERENCES

[1] D. Borthakur, "HDFS Architecture Guide," *Hadoop Apache Project*, 2008. [Online]. Available: http://hadoop.apache.org/common/docs/current/hdfs_design.pdf

[2] B. Calder, J. Wang, A. Ogus, N. Nilakantan, A. Skjolsvold, S. McKelvie, Y. Xu, S. Srivastav, J. Wu, H. Simitci, J. Haridas, C. Uddaraju, H. Khatri, A. Edwards, V. Bedekar, S. Mainali, R. Abbasi, A. Agarwal, M. F. ul Haq, M. I. ul Haq, D. Bhardwaj, S. Dayanand, A. Adusumilli, M. McNett, S. Sankaran, K. Manivannan, and L. Rigas, "Windows Azure Storage: A Highly Available Cloud Storage Service with Strong Consistency," in *Proc. of ACM Symposium on Operating Systems Principles (SOSP)*, 2011.

[3] C. Huang, H. Simitci, Y. Xu, A. Ogus, B. Calder, P. Gopalan, J. Li, and S. Yekhanin, "Erasure Coding in Windows Azure Storage," in *Proc. USENIX Annual Technical Conference (USENIX ATC)*, 2012.

[4] M. Sathiamoorthy, M. Asteris, D. Papailiopoulos, A. G. Dimakis, R. Vadali, S. Chen, and D. Borthakur, "XORing Elephants: Novel Erasure Codes for Big Data," *Proc. VLDB Endowment (to appear)*, 2013.

[5] I. Reed and G. Solomon, "Polynomial Codes over Certain Finite Fields," *Journal of the Society for Industrial and Applied Mathematics*, vol. 8, no. 2, pp. 300–304, 1960.

[6] K. V. Rashmi, N. B. Shah, D. Gu, H. Kuang, D. Borthakur, and K. Ramchandran, "A Solution to the Network Challenges of Data Recovery in Erasure-coded Distributed Storage Systems: A Study on the Facebook Warehouse Cluster," in *Proc. 5th USENIX Workshop on Hot Topics in Storage and File Systems*, 2013.

[7] A. G. Dimakis, P. B. Godfrey, Y. Wu, M. J. Wainwright, and K. Ramchandran, "Network Coding for Distributed Storage Systems," *IEEE Trans. Inform. Theory*, vol. 56, no. 9, pp. 4539–4551, 2010.

[8] R. Bhagwan, K. Tati, Y.-C. Cheng, S. Savage, and G. M. Voelker, "Total Recall: System Support for Automated Availability Management," in *Proc. USENIX Symposium on Networked Systems Design and Implementation (NSDI)*, 2004.

[9] Y. Hu, Y. Xu, X. Wang, C. Zhan, and P. Li, "Cooperative Recovery of Distributed Storage Systems from Multiple Losses with Network Coding," *IEEE Journal on Selected Areas in Communications*, vol. 28, no. 2, pp. 268–276, 2010.

[10] K. W. Shum, "Cooperative Regenerating Codes for Distributed Storage Systems," in *Proc. IEEE International Conference on Communications (ICC)*, 2011.

[11] A. Wang and Z. Zhang, "Exact Cooperative Regenerating Codes with Minimum-Repair-Bandwidth for Distributed Storage," in *Proc. IEEE INFOCOM*, 2013.

[12] 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 Trans. Inform. Theory*, vol. 57, no. 8, pp. 5227–5239, 2011.

[13] Y. Wu and A. Dimakis, "Reducing Repair Traffic for Erasure Coding-based Storage via Interference Alignment," in *Proc. IEEE Int. Symp. Inform. Theory (ISIT)*, 2009.

[14] Y. Wu, A. G. Dimakis, and K. Ramchandran, "Deterministic Regenerating Codes for Distributed Storage," *Allerton Conference on Control, Computing, and Communication*, 2007.

[15] Y. Wu, "Existence and Construction of Capacity-Achieving Network Codes for Distributed Storage," *IEEE Journal on Selected Areas in Communications*, vol. 28, no. 2, pp. 277–288, 2010.

[16] V. R. Cadambe, S. A. Jafar, and H. Maleki, "Distributed Data Storage with Minimum Storage Regenerating Codes - Exact and Functional Repair are Asymptotically Equally Efficient," in *Proc. 2010 Wireless Network Coding (WINC) Workshop*, 2010.

[17] N. B. Shah, K. V. Rashmi, P. V. Kumar, and K. Ramchandran, "Interference Alignment in Regenerating Codes for Distributed Storage: Necessity and Code Constructions," *IEEE Trans. on Inform. Theory*, vol. 58, no. 4, pp. 2134–2158, 2012.

[18] N. B. Shah, K. V. Rashmi, P. Vijay Kumar, and K. Ramchandran, "Distributed Storage Codes With Repair-by-Transfer and Nonachievability of Interior Points on the Storage-Bandwidth Tradeoff," *IEEE Trans. on Inform. Theory,*, vol. 58, no. 3, pp. 1837–1852, 2012.

[19] A. Kermarrec and N. Le, "Repairing Multiple Failures with Coordinated and Adaptive Regenerating Codes," *IEEE International Symposium on Network Coding (NetCod)*, 2011.

[20] N. Le Scouarnec, "Exact Scalar Minimum Storage Coordinated Regenerating Codes," *Proc. IEEE International Symposium on Information Theory (ISIT)*, 2012.

[21] J. Chen and K. W. Shum, "Repairing Multiple Failures in the Suh-Ramchandran Regenerating Codes," in *Proc. IEEE Int. Symp. Inform. Theory (ISIT)*, 2013.

[22] T. Andreescu and R. Gelca, *Mathematical Olympiad Challenges*. Springer, 2000.

[23] Y. S. Han, H.-T. Pai, R. Zheng, and P. K. Varshney, "Update-Efficient Regenerating Codes with Minimum Per-Node Storage," in *Proc. IEEE Int. Symp. Inform. Theory (ISIT)*, 2013.