Two-layer Locally Repairable Codes for Distributed Storage Systems

Hongmei Xie and Zhiyuan Yan
Department of Electrical and Computer Engineering, Lehigh University, Bethlehem, PA 18015, USA
E-mail: {hox209; yan}@lehigh.edu

Abstract—In this paper, we propose locally repairable codes (LRCs) with optimal minimum distance for distributed storage systems (DSS). A two-layer encoding structure is employed to ensure data reconstruction and the designated repair locality. The data is first encoded in the first layer by any existing maximum distance separable (MDS) codes, and then the encoded symbols are divided into non-overlapping groups and encoded by an MDS array code in the second layer. The encoding in the second layer provides enough redundancy for local repair, while the overall code performs recovery of the data based on redundancy from both layers. Our codes can be constructed over a finite field with size growing linearly with the total number of nodes in the DSS, and facilitate efficient degraded reads.

Index Terms—Distributed storage system, local repair, optimal minimum distance

I. INTRODUCTION

In distributed storage systems (DSSs), data is spread across nodes in the network, while users are geographically dispersed. To avoid data loss from node failures, coding techniques such as replication and erasure codes [1] are employed to create redundancy for two types of data recovery [2]. First, a user (data collector) must be able to retrieve the original data by contacting a certain number of storage nodes in the system, called data reconstruction. Second, data stored in a failed node should be recovered and stored in a new replacement node by contacting other nodes in the network, called data repair.

In larger scale DSSs, such as cloud and peer-to-peer storage systems, node failure is a routine rather than an exception, hence the repair problem attracts much interest in recent literature. Different metrics have been introduced to measure the cost of data repair, such as repair bandwidth [3], the total amount of information downloaded from exiting nodes, and repair locality [4]–[7], the number of nodes contacted by the replacement node in the repair process. As pointed out in [7], the number of nodes involved in data repair is closely related to the disk input/output (I/O) overhead, which is the main performance bottleneck in the repair problem.

Various codes have been proposed to reduce repair locality, such as scalar linear codes [6] [8] and vector codes [7] [9] [10]. In [6] [8], extra parity constraints are introduced into encoded symbols of an MDS code to enhance repair locality, and a trade-off is demonstrated between the minimum distance and the repair locality of the resulting code. The trade-off is extended in [7] to accommodate vector codes for local repair of one failed node, and explicit code construction based on a two-layer encoding structure is proposed for a specific set of parameters. Vector codes capable of repairing more than one failed node locally at the same time are proposed in [10]. Those vector codes with local repair property are called locally repairable codes (LRCs), and LRCs achieving the optimal minimum distance are said to be optimal. Optimal LRCs are also constructed in [10] featuring a two-layer encoding structure, where Gabidulin codes [11] are used in the first layer encoding. However, the adoption of Gabidulin codes leads to a finite field size growing exponentially with the number of nodes in the DSS.

In this paper, we construct optimal LRCs with a similar two-layer encoding structure, but prove that any MDS codes can be used in the first layer to ensure data reconstruction, if the MDS array code in the second layer provides enough redundancy for local repair. Naturally, the construction in [10] can be viewed as a special case of ours, as Gabidulin codes are also a family of MDS codes. Compared to [10], our code is guaranteed to exist in a finite field with size proportional to the total number of nodes in the DSS. Our construction has flexible structures, and leads to codes proposed in [7] given the same set of parameters. However, compared to the construction in [7], our codes have smaller penalty when successive reads are performed in the scenario of degraded reads.

It should be pointed out that the two-layer encoding (or concatenated encoding) is not a new technique. It has been widely used in the literature for different desired advantages in the DSS aside from [7] [10]. For example, fractional repetition codes are proposed in [12] from concatenation of an outer MDS code and an inner repetition code for uncoded repair process. An outer MDS code and an inner fractional repetition code are employed in [13] to construct regenerating codes with local, exact and uncoded repair. Scalar linear codes such as Pyramid codes investigated in [6] [14] and that proposed in [8] can also be viewed as examples of two-layer encoding.

The rest of the paper is organized as follows. Section II provides some preliminaries of repair locality in DSS, followed by our LRC construction in Section III. Section IV concludes the paper.

II. LOCALLY REPAIRABLE CODES

Suppose there are \( n \) nodes in a DSS, each having a storage capacity of \( \alpha \). Given a message file of \( M \) symbols over GF\( (q) \) \( (q \) is a prime power), a vector code \( C \) encodes the \( M \) symbols
into a codeword \( c = [c_0 c_1 \ldots c_{n-1}] \), where \( c_i \in \text{GF}(q)^r \) is a column vector over \( \text{GF}(q) \) and represents the encoded symbols stored in node \( i \) for \( i \in [n] \). Throughout this paper, \([n]\) denotes the set \( \{0, 1, \ldots, n-1\} \). The minimum distance \( d \) of \( C \) is defined to be the minimum number of erased nodes so that the entropy of the non-erased coded nodes is strictly less than \( M \) \([7] [10]\). That is,

\[
d = n - \max_{S: H(c_S) \leq M} |S|,
\]

where \( S \subseteq [n] \) and \( c_S = \{c_i : i \in S\} \). The code \( C \) is referred to as an \((n, \alpha, M, d)\) vector code.

If for each coded symbol \( c_i \), with \( i \in [n] \) of a codeword \( c \in C \), there exists a set of nodes \( \Gamma(i) \subseteq [n] \) such that 1) \( i \in \Gamma(i) \); 2) \( |\Gamma(i)| \leq r + \delta - 1 \); and 3) minimum distance of \( C|\Gamma(i) \) is at least \( \delta \), where \( r, \delta \) are positive integers and \( C|\Gamma(i) \) is the code obtained by restricting \( C \) over \( \Gamma(i) \), then \( C \) is said to have \((r, \delta)\) locality \([10]\). Note that the \((r, \delta)\) locality indicates that each node \( i \in [n] \) can be expressed as a function of at most \( r \) other elements \( j \in \Gamma(i)\backslash\{i\} \), a property called locally repairable, and \( \Gamma(i) \) is referred to as a local repair group. The \((n, \alpha, M, d)\) vector code \( C \) is then called a locally repairable code (LRC) \([7] [10]\), denoted as \((n, \alpha, M, d; r, \delta)\) LRC \( C \).

It is established in \([15]\) that the minimum distance of an \((n, \alpha, M, d; r, \delta)\) LRC code is bounded by

\[
d \leq n - \left\lceil \frac{M}{\alpha} \right\rceil + 1 - \left( \left\lceil \frac{M}{r\alpha} \right\rceil - 1 \right)(\delta - 1),
\]

and codes attaining this bound are said to be optimal. When \( \delta \) is fixed at 2, the bound in Eq. (2) reduces to \( d \leq n - \left\lceil \frac{M}{\alpha} \right\rceil - \left\lceil \frac{M}{r\alpha} \right\rceil + 2 \), which was first proved in \([7]\).

The upper bound in Eq. (2) is proved in \([15]\), based on an iterative algorithm that finds a set \( S \) in Eq. (1) in a fast way, and bounds the minimum distance \( d \) accordingly. Generally speaking, in each iteration, the algorithm picks a node and adds its local repair group into the current set \( S \). If this group has at least \( d - 1 \) nodes outside the current \( S \), then \( S \) is updated to accommodate the newly added nodes. The iteration carries on till the set \( S \) provides entropy \( \left\lceil \frac{M}{\alpha} \right\rceil \alpha - \alpha \).

A two-layer encoding scheme is used in \([10]\) to construct LRC codes that reach the optimal minimum distance in Eq. (2), based on Gabidulin codes \([11]\) and MDS array codes. When \( \delta = 2 \), a similar two-layer encoding approach is also proposed to obtain optimal LRC codes with parameters that satisfy \( (r + 1)n \) and \( r + 1 = \alpha \) in \([7]\). Though codes in \([10]\) work for general parameter settings, they are constructed in a finite field whose size grows exponentially with \( n\alpha \), leading to high computational complexities in data reconstruction and repair.

### III. General Code Construction

In this paper, we propose optimal LRC codes based on the same two-layer encoding structure as in \([10]\), but prove that any MDS code suffices to obtain the optimal minimum distance. As a result, the construction in \([10]\) can be viewed as a special case of our scheme, as Gabidulin codes are also MDS codes. However, our construction leads to optimal LRC code over a finite field with size growing linearly with respect to \( n\alpha \), when other MDS codes such as Reed-Solomon (RS) codes are used.

It should be pointed out that although we use the same two-layer encoding structure as that in \([10]\), we tackle the problem from a different perspective. In \([10]\), Gabidulin codes are considered so that \( d - 1 \) node erasures can be turned into equivalent rank erasures, which are then proved to be correctable by the Gabidulin code. In our approach, we rely on the redundancy provided by the first-layer MDS code, and show that \( n - d + 1 \) nodes provide sufficient entropy of the original data.

First we discuss the achievability of the optimal distance in Eq. (2) under different parameter settings, and then present our code construction accordingly.

#### A. Achievability of Optimal Distance

As mentioned in Section II, an iterative algorithm is used in \([15]\) to prove the upper bound in Eq. (2). Based on the work in \([15]\), we further claim the achievability of the optimal distance in Lemma 1.

**Lemma 1.** The optimal distance in Eq. (2) is reachable if and only if \( (r + \delta - 1)n \), or \( n((r + \delta - 1) - (\delta - 1)) \geq \left\lceil \frac{M}{\alpha} \right\rceil \mod r > 0 \) when \( (r + \delta - 1) \nmid n \).

**Proof:** The optimal distance \( d \) in Eq. (2) can be obtained in two cases by using the algorithm in \([15]\).

- The algorithm in \([15]\) terminates in \( \left\lceil \frac{M}{\alpha} \right\rceil - 1 \) steps, with each step adding exactly \( r + \delta - 1 \) nodes with entropy \( r\alpha \). Hence local repair groups should be non-overlapping in the first place. Further, the algorithm should reach the same result regardless of which \( \left\lceil \frac{M}{\alpha} \right\rceil - 1 \) local repair groups are selected. Hence \( (r + \delta - 1)n \). Simply speaking, the optimal distance is achievable only if nodes in the DSS can be divided into non-overlapping local repair groups of the same size. Conversely, if we can divide the nodes into non-overlapping groups of size \( r + \delta - 1 \), any \((r, \delta)\) LRC codes can be used to accommodate entropy of \( r\alpha \) within each group, which satisfies the termination of the algorithm.

- The algorithm in \([15]\) terminates in \( \left\lceil \frac{M}{\alpha} \right\rceil \) steps, where \( r + \delta - 1 \) nodes with entropy \( r\alpha \) are added in each of the first \( \left\lceil \frac{M}{\alpha} \right\rceil - 1 \) steps, and other nodes with entropy \( (\alpha - \delta + 1)\alpha \) are added in the last step, where \( \alpha \) is a positive integer. As in the first case, non-overlapping local repair groups are also required. The last portion of entropy indicates that \( n((r + \delta - 1) - (\delta - 1)) \geq \left\lceil \frac{M}{\alpha} \right\rceil \mod r > 0 \) for the last group added to provide this portion of entropy. In other words, when \( (r + 1) \nmid n \), the optimal distance \( d \) is reachable only if \( n((r + \delta - 1) - (\delta - 1)) \geq \left\lceil \frac{M}{\alpha} \right\rceil \mod r > 0 \). The converse can be proved similarly as in the first case.

#### B. Code Structure

We use the same two-layer encoding structure as that in \([10]\) to construct optimal LRCs for the two cases in Lemma 1.
scalar MDS code is used in the first layer, whose encoded symbols are partitioned into sets of size \( r_0 \alpha \), to be stored in non-overlapping local repair groups. Then a second layer encoding is performed within each local repair group by an \((r + \delta - 1, r)\) MDS array code to ensure \((r, \delta)\) locality. The overall code reaches the desired repair locality as well as the optimal minimum distance.

Suppose a message file of \( M \) symbols over GF\((q)\) is to be encoded and stored in a DSS with \( n \) nodes, each of which has a storage capacity of \( \alpha \) symbols. We construct \((n, \alpha, M, d; r, \delta)\) LRC codes such that the system has \((r, \delta)\) locality and minimum distance \( d = n - \left\lceil \frac{M}{\alpha} \right\rceil + 1 - \left\lfloor \left( \frac{M}{\alpha} \right) - 1 \right\rfloor (\delta - 1) \), or any \( k^* = \left\lceil \frac{M}{\alpha} \right\rceil + \left\lfloor \left( \frac{M}{\alpha} \right) - 1 \right\rfloor (\delta - 1) \) nodes suffice for data reconstruction. Note that if \( M \leq r_0 \alpha \), each node will be repaired locally by \( r \) other nodes, while the traditional repair through data reconstruction needs \( \left\lceil \frac{M}{\alpha} \right\rceil \leq r \) nodes, which is not the purpose of LRC codes. Hence we assume \( M > r_0 \alpha \) in the rest of this paper. We introduce \( M^* = \left\lceil \frac{M}{\alpha} \right\rceil \alpha \) for simplicity. It can be easily shown that \( \left\lceil \frac{M}{\alpha} \right\rceil = \left\lceil \frac{M^*}{\alpha} \right\rceil \) and \( \left\lfloor \left( \frac{M}{\alpha} \right) - 1 \right\rfloor = \left\lfloor \left( \frac{M^*}{\alpha} \right) - 1 \right\rfloor \), hence \( k^* = \frac{M^*}{\alpha} + \left( \frac{M^*}{\alpha} - 1 \right) (\delta - 1) \).

As discussed in Section III-A, we assume non-overlapping local repair groups to obtain optimal distance. Suppose the \( n \) nodes of storage capacity \( \alpha \) are labeled as \( 0, 1, \ldots, n \), and divided into \( \Delta \) non-overlapping groups, denoted by \( G_j \), where \( \Delta \) is a positive integer greater than 1, and \( j \in \{\Delta\} \). Depending on whether \( r + \delta - 1 \) divides \( n \) or not, we construct the optimal LRC code for the two cases accordingly.

1) \( n = \Delta (r + \delta - 1) \). The optimal distance achievable analysis in Section III-A suggests that \( \delta - 1 \) nodes in each local repair group will solely be used for local repair, while the other \( r \) nodes ensures reliability of the overall code. Hence we need to embed \( M \) information symbols in \( \Delta r_0 \alpha \) encoded symbols. First we pad \( M^* - M \) zeros into the message, and use a \((\Delta r_0, M^*)\) MDS code \( C^\dagger \) to obtain \( \Delta r_0 \alpha \) encoded symbols, and store them into the first \( r_0 \) nodes of each repair group, shown as the blank areas in Fig. 1, with \( r_{\Delta-1} = r \). Based on Fig. 1 we may refer to a node as a column, and another dimension a row of the DSS. Next, within each group \( G_j \), an \((r + \delta - 1, r)\) systematic MDS array code \( C^\dagger \alpha \) over GF\((q)\) is used to encode the \( r_0 \alpha \) encoded symbols of \( C^\dagger \) and store the parity checks in the \( \delta - 1 \) shaded columns.

2) \( n = \Delta (r + \delta - 1) + r_{\Delta-1} + \delta - 1 \). where \( r_{\Delta-1} \geq \left\lceil \frac{M^*}{\alpha} \right\rceil (\text{mod} \ r) > 0 \). Similarly, we pad zeros to the message file if necessary, and encode \( M^* \) symbols into \((\Delta - 1) r_0 + r_{\Delta-1} \alpha \) symbols using a \((\Delta - 1) r_0 + r_{\Delta-1} \alpha, M^*\) MDS code \( C^\dagger \alpha \), and store them in the blank columns of Fig. 1. Next, each of the first \( \Delta - 1 \) groups employs an \((r + \delta - 1, r)\) systematic MDS array code \( C^\dagger \alpha \) over GF\((q)\), and store the parity checks in its shaded columns respectively, as in the first case. For \( G_{\Delta-1} \), an \((r_{\Delta-1} + \delta - 1, r_{\Delta-1})\) systematic MDS array code \( C^\dagger \alpha \) over GF\((q)\) is used to obtain coded symbols in the last \( r_{\Delta-1} + \delta - 1 \) nodes.

In both cases, we obtain an \((n, \alpha, M^*)\) vector code \( C \). Let \( m = (m_0 m_1 \ldots m_{M-1})^T \) be the message vector after padding of zeros (if necessary), where \( T \) is the transpose operation. Suppose \( c = (c_{i,j})^T = (g_i^T m)^T \) is the corresponding codeword, where \( c_{i,j} \) is the coded symbol stored in the \( j \)th row of node (column) \( i \), and \( g_i \) a column vector called a generator vector for \( i \in [n] \) and \( j \in [\alpha] \). We can write a generator matrix \( G \) for \( C \) to be

\[
G = \begin{bmatrix}
(g_{0,0} g_{1,0} \cdots g_{0,\alpha-1} g_{1,0} \cdots g_{1,\alpha-1} \cdots g_{n-1,\alpha-1})^T
\end{bmatrix},
\]

and define \( G_i = (g_{i,0} g_{i,1} \cdots g_{i,\alpha-1}) \), called a node generator for \( i \in [n] \).

Note that \( C^\dagger \alpha \) (and \( C^\dagger \)) in case 2) is an \((r + \delta - 1, r)\) systematic MDS array code over GF\((q)\), which can be obtained by employing a systematic \((r + \delta - 1, r)\) MDS scalar code over GF\((q)\) in each row of the local repair group for simplicity. In this case, we get \( G^\dagger (1) = (g_{j,j}) \) where \( j \in [\alpha] \) and \( i' \in [n] \setminus \{t(r + \delta - 1) + r_0, t(r + \delta - 1) + r_0 + 1, \ldots, t(r + \delta - 1) + r_0 + \delta - 2 : r_0 \in \{r, r_{\Delta-1}\}, t \in [\Delta - 1]\} \). Then \( G^\dagger (1) \) is a generator matrix of \( C^\dagger (1) \). Let \( G^\dagger (2) = (I_0, I_1, \ldots, I_{r_0-1}, \eta_0, \eta_1, \ldots, \eta_{\alpha-2}) \) be a generator matrix of \( C^\dagger (2) \) (or \( C^\dagger (3) \)), where \( I_j, j \in [r_0] \) is the \( j \)th column of the \( r_0 \times r_0 \) identity matrix \( I_{r_0} \). Then

\[
g_{i+r_0+\epsilon, \ell} = [g_{i,\epsilon} g_{i+1,\epsilon} \cdots g_{i+r_{\Delta-1},\epsilon}] \eta_\epsilon, \quad (3)
\]

where \( i = j(r + \delta - 1) \) for \( j \in [\Delta - 1], \epsilon \in [\delta - 1], \ell \in [\alpha] \), and \( r_0 \in \{r, r_{\Delta-1}\} \). Eq. (3) establishes the linear dependency of generator vectors of \( C^\dagger (2) \) and \( C^\dagger (3) \) on that of \( C^\dagger (1) \). A different set of equations will be obtained if we use arbitrary \((r + \delta - 1, r)\) systematic MDS array code over GF\((q)\) for \( C^\dagger (2) \) \((C^\dagger (3))\), but the dependency between generator vectors of \( C^\dagger (2) \) and \( C^\dagger (3) \) on that of \( C^\dagger (1) \) will not be changed, which is the basis of our proof of the data reconstruction.

C. Local Repair and Data Reconstruction

As described in the previous section, the local repair property is solely determined by the second layer encoding, which guarantees the desired locality for the DSS.

Theorem 1. The LRC code constructed in Section III has a repair locality of \( r \).

Proof: Each node participates in a \((r + \delta - 1, r)\) or \((r_{\Delta-1} + \delta - 1, r_{\Delta-1})\) MDS array code, which can be repaired by at most \( r \) other nodes. Hence the LRC code has a repair locality of \( r \).

The data reconstruction, on the other hand, depends on the erasure correction capability of the overall code \( C \). The problem requires the original message be recovered based on

![Fig. 1. Two-layer encoding structure.](image-url)
any \( k^* \) nodes, or equivalently, \( k^* \alpha \) generator vectors from any \( k^* \) node generators have rank \( M^* \). We will show that this requirement can be fulfilled by any MDS code \( C^{(1)} \), relying on the following fact of MDS codes.

**Fact 1.** If \( S \) is a subset of generator vectors of an \((N,K)\) MDS code, then its elements are linearly independent as long as \(|S| \leq K\).

Note that generator vectors from blank columns of Fig. 1 are exactly the same ones from \( C^{(1)} \), hence any subset of them have full rank as long as the set size is no greater than \( M^* \). In particular, those from the same local repair group \( G_j \) span a vector space \( V_j \) of dimension \( r_0 \alpha \) over \( GF(q) \) given that \( r_\alpha < M \leq M^* \), where \( j \in [\Delta] \), \( r_0 \in \{r,r_{\Delta-1}\} \). Based on the same logic, we can have the following claim.

**Claim 1.** Subspaces of \( V_j \)’s have trivial intersections if the summation of their dimensions is no greater than \( M^* \).

Meanwhile, generator vectors in the shaded columns are linear combinations of that from the blank columns, as shown in Eq. (5). Hence all the \( r_0 + \delta - 1 \) nodes in \( G_j \) span the same vector space \( V_j \) as obtained from the first \( r_0 \) nodes for \( j \in [\Delta] \), \( r_0 \in \{r,r_{\Delta-1}\} \), as shown in Fig. 1. Now we are ready to prove the data reconstruction of \( C \).

**Theorem 2.** Data reconstruction can be performed by any \( k^* = \lceil \frac{M^*}{\alpha} \rceil + (\lceil \frac{M^*}{\alpha} \rceil - 1)(\delta - 1) \) columns of code \( C \).

**Proof:** The theorem can be proved by showing that the \( k^* \alpha \) generator vectors corresponding to any \( k^* \) columns of \( C \) span a vector space of dimension \( M^* \) over \( GF(q) \). We prove this for the two cases of construction in Section III respectively.

1) \( n = \Delta(r+\delta-1) \). From previous analysis, subspaces with smaller dimensions can be obtained by picking nodes from the same local repair group as many as possible. Since \( \alpha | M^* \), we can write \( M^* = \lambda \alpha + r_1 \alpha \) where \( 0 \leq r_1 \leq r-1 \), and have two different \( k^* \)'s:

- \( k^* = \lambda r + (\lambda - 1)(\delta - 1) = (\lambda - 1)(r + \delta - 1) + r \) if \( r_1 = 0 \) or \( r_\alpha | M^* \). In this case, we pick the \( k^* \) nodes by first choosing \( (\lambda - 1)(r + \delta - 1) \) nodes from \( \lambda - 1 \) different local repair groups, and then selecting another \( r \) nodes randomly from the remaining repair groups. Based on Claim 1, the last \( r \) nodes span a subspace \( U_0 \) of dimension \( r_\alpha \), while the first \( (\lambda - 1)(r + \delta - 1) \) nodes span a subspace \( U_1 \) of dimension \( (\lambda - 1)r_\alpha \) given that \( r_\alpha < M^* \) and \( (\lambda - 1)r_\alpha < M^* \). Using Claim 1 again, \( U_0 \) and \( U_1 \) have only trivial intersection since the summation of their dimensions is exactly \( M^* \). Hence any \( k^* \) nodes will span a subspace with dimension at least \( M^* \).

- \( k^* = \lambda (r + \delta - 1) + r_1 \) if \( r_1 > 0 \) or \( r_\alpha \nmid M^* \). Similarly, we compose the \( k^* \) worst-case nodes with \( \lambda (r + \delta - 1) \) elements from \( \lambda \) different local groups, and \( r_1 \) others randomly selected from the remaining groups. Following a similar argument as the first case, those nodes span a vector space of dimension \( \lambda \alpha + r_1 \alpha = M^* \).

In either setting, any \( k^* \) nodes will span a subspace with dimension at least \( M^* \). Conversely, the largest possible dimension spanned by any \( k^* \) nodes is also \( M^* \). Hence generator vectors from any \( k^* = \lceil \frac{M^*}{\alpha} \rceil + (\lceil \frac{M^*}{\alpha} \rceil - 1)(\delta - 1) \) nodes span an \( M^* \)-dimensional vector space over \( GF(q) \), and the original message file can be reconstructed.

**Example 1.** We construct an example with \( n = 6, \alpha = 3, M = 8, r = 2, \delta = 2 \), leading to \( M^* = 9, k^* = 4 \) and a designed distance \( d = 3 \). A \((12,9)\) RS code over \( GF(2^4) \) is adopted as \( C^{(1)} \), and \( C^{(2)} a \) \((3,2)\) single parity check code (an MDS code).

A length-8 message is first padded with a 0. Suppose \( c = (c_{i,j})^T \in C^{(1)} \) is obtained in the first layer encoding, where \( i \in \{0,1,3,4\}, j \in \{0,1,2\} \). The corresponding codeword of \( C \) is shown in Table 7 where \( v_i \) is storage node \( i \) with \( i \in [6] \), and \( c_{i,j} = c_{i-1,j} + c_{i-2,j} \) for \( i \in \{2,5\}, j \in \{0,1,2\} \). It can be verified that any node has a repair locality of 2. Now we try to reconstruct the message from the four nodes 0, 1, 2, 5. Apparently node 2 is totally redundant, hence we form the following 9 equations,

\[
\begin{pmatrix}
    c_{0,0} \\
    c_{0,1} \\
    c_{1,2} \\
    c_{5,0} \\
    c_{5,1} \\
    c_{5,2} \\
\end{pmatrix} = 
\begin{pmatrix}
    g_{0,0}^T \\
    g_{0,1}^T \\
    g_{1,2}^T \\
    g_{5,0}^T \\
    g_{5,1}^T \\
    g_{5,2}^T \\
\end{pmatrix} \begin{pmatrix}
    m_0 \\
    m_1 \\
    m_2 \\
    m_3 \\
    m_4 \\
    m_5 \\
\end{pmatrix},
\]

where \( I_6 \) is the 6-by-6 identity matrix, and

\[
B = \begin{pmatrix}
    1 & 0 & 0 & 1 & 0 & 0 \\
    0 & 1 & 0 & 0 & 1 & 0 \\
    0 & 0 & 1 & 0 & 0 & 1 \\
\end{pmatrix}.
\]

It is easy to check that both matrices on the right hand side of Eq. 4 have rank 9, and their product is invertible. Hence the message vector can be solved from the matrix equation. It can be verified that any \( k^* = 4 \) nodes suffice to reconstruct \( m \), or the optimal minimum distance \( d = 3 \) is reached, hence the code in Table 7 is a \((6,3,8;3;2,3)\) optimal LRC code.

**D. Relation to Other Works**

The general construction in Section III requires \( C^{(1)} \) to be an MDS code, and any existing MDS code, such as Reed-Solomon codes, Gabidulin codes, new MDS codes or even MSR codes for distributed storage, can be
used. In particular, if a Gabidulin code is used, we obtain codes proposed in [10]. Hence the codes in [10] can be viewed as a special case of our construction. However, the adoption of Gabidulin codes requires a field size of $q^{M^r}$, basically exponential with respect to $na$. On the other hand, if RS codes are used for $C^{(1)}$, our code can be constructed over a finite field of size $q > na$, linear with respect to the total number of nodes in the DSS.

Furthermore, we can also use regenerating codes such as MSR codes in each local repair group as in [10], and reduce bandwidth during local repair process. Suppose $r_d$ nodes are to be connected within each local group for the repair, each of which sends $\beta$ symbols. It is proved in [10] that the optimum bandwidth has $\beta = \alpha/(r_d - r + 1)$, and the size of the original file is upper bounded by

$$M \leq \left\lfloor \frac{n - d + 1}{r + \delta - 1} \right\rfloor \alpha + \min\{h, r\} \alpha,$$

where $h = n - d + 1 - (r + \delta - 1) \left\lfloor (n - d + 1)/(r + \delta - 1) \right\rfloor$.

Optimal LRC codes are also constructed in [7] for $\delta = 2$, $(r + 1)n$ and $r + 1 = \alpha$. Our construction is flexible, and leads to the codes in [7] under the same set of parameters.

Our proof of Theorem 2 is based on subspaces spanned by generator vectors of $C^{(1)}$. Note that the same $V_j$ will be obtained if we rearrange coded symbols (generator vectors) of $C^{(1)}$ within each local repair group in Fig.1 or use a different MDS code $C^{(2)}$ in the second layer encoding. In particular, for $r + 1 = \alpha$, we can store the $r\alpha$ encoded symbols of $C^{(1)}$ in the first $r$ rows, and the $\alpha$ parity checks from $C^{(2)}$ in the last row, instead of the last column of $G_j$.

Note that encoded symbols of the same codeword of $C^{(2)}$ should come from different columns (nodes) in order to obtain the desired repair locality, which can be implemented by simple permutations. Let $\pi = (r, r - 1, \ldots, 0)$, and $\pi_{\ell}$ the $\ell$-th right circulant of $\pi$, that is, $\pi_{\ell}(t) = \pi((t + \ell) \bmod (r + 1))$ for $\ell, t \in [r + 1]$. Then the $\ell$-th codeword of $C^{(2)}$ is obtained by $\sum_{t=0}^{r} \xi_{t, i} = 0$, where $i = \pi_{\ell}(t)$. For example, $\pi_r = (r - 1, r - 2, \ldots, 0, r)$, and $c_{r-1,0} + c_{r-2,1} + \cdots + c_{r, r} = 0$, from which $c_{r, r}$ can be calculated and stored into row $r$ of node $r$.

Table I gives another $(6, 3, 8, 3; 2, 3)$ optimal LRC code with exactly the same parameters as that in Table II. Note that given the same input message, codeword $b = (b_0, b_1, \ldots, b_{11})^T$ in Table II is the same as $c = (c_{i,j})^T$ for $i \in \{0, 1, 3, 4\}$, $j \in \{0, 1, 2\}$ in Table I. However, parity check symbols from $C^{(2)}$ are formed according to the permutation approach above to ensure a repair locality of 2.

Our construction uses a $(12, 9)$ MDS code $C^{(1)}$ to obtain coded symbols in the first two rows, which ensures recovery of no more than two erasures. In this special case, we can also use a $(6, 4)$ MDS code in each row, leading to a code presented in [7].

### Table I

| $n_0$ | $n_1$ | $n_2$ | $n_3$ | $n_4$ | $n_5$ |
|-------|-------|-------|-------|-------|-------|
| $c_{0,0}, c_{0,1}, c_{0,2}$ | $c_{0,0} + c_{1,0}, c_{0,1} + c_{1,1}, c_{0,2} + c_{1,2}$ | $c_{3,0}, c_{4,0}, c_{3,0} + c_{4,0}$ | $c_{3,1}, c_{4,1}, c_{3,1} + c_{4,1}$ | $c_{3,2}, c_{4,2}, c_{3,2} + c_{4,2}$ | $c_{3,3}, c_{4,3}, c_{3,3} + c_{4,3}$ |

### Table II

| $n_0$ | $n_1$ | $n_2$ | $n_3$ | $n_4$ | $n_5$ |
|-------|-------|-------|-------|-------|-------|
| $b_0, b_1, b_2, b_3, b_4$ | $b_5, b_6, b_7, b_8, b_9$ | $b_{10}, b_{11}, b_{12}, b_{13}, b_{14}$ | $b_{15}, b_{16}, b_{17}, b_{18}, b_{19}$ | $b_{20}, b_{21}, b_{22}, b_{23}, b_{24}$ |

E. Degraded Reads

Both schemes in Table I and II achieve the desired data repair and reconstruction parameters. However, compared to [7], our codes in Section III-B feature other merits such as efficient degraded reads [21].

As pointed out in [21] and the references therein, disk failures are dominated by temporary unavailability due to network partitions, software updates and so on. In the period between failure and recovery, reads are degraded because data from failed nodes must be recovered to complete the read process. For single disk failures, a penalty is defined to be the number of symbols required to perform the read minus the number of symbols desired to be read.

If random reads do not cause extra cost (e.g., delay), both codes from our construction and [7] induce a penalty of at most $r - 1$ in degraded reads. Given the same repair locality of $r$, reading one symbol from a failed node can be performed by reading at most $r$ other symbols in other nodes. In practice, however, reading from random positions of a disk could be time consuming, and successive reads are preferred. In this case, degraded reads performed by our construction have less penalty than that in [7].

Suppose a systematic code $C$ is used for simplicity, and the message symbols are stored in data disks $D_0, D_1, \ldots, D_{s-1}$, and parity symbols are stored in parity disks $P_0, P_1, \ldots, P_r$, respectively, as shown in Fig.2 (reproduced from [21] Fig.1), where $s = \lceil \frac{M}{r} \rceil$ and $t = n - s - 1$. Without loss of generality, let us assume the first $\Delta$ parity disks store the single parity check symbols of $C^{(2)}$ for local group 0, 1, $\Delta - 1$, respectively, and the rest stores that from $C^{(1)}$.
As in [21], we assume contiguous data symbols are stored in successive disks to take better advantage of parallel I/O, i.e., successive reads are performed from the starting point to the end in a row by row manner. For our construction, at most \( r \) extra symbols in the same row are required to be read if one node fails, as local repair constraints are conducted row wisely. Hence a penalty of at most \( r - 1 \) is necessary. On the other hand, the structure in [7] stores the \( \alpha = r + 1 \) symbols participating in the same parity check equation of \( C(2) \) in different rows. To be specific, node \( t \) stores a symbol of the \( t \)-th codeword in row \( \pi(t) \), hence up to \( \pi(t) + 1 \) symbols are to be read from node \( t \) to repair some other symbol participating in the same codeword. Given that \( \pi(t) \in [r + 1] \), in the worst case, reading of \( r + 1, r, \ldots, 2 \) symbols (rows) from \( r \) nodes respectively is necessary to repair one symbol in a failed node. Hence a penalty of \((r+1)(r+2)/2 - 2\) is resulted, in the order of \( O(r^2) \).

For example, suppose node 0 fails in Table [I] and \( b_0 \) is to be read. If using successive reads, we have to read 3 symbols in node \( n_1 \) till \( b_0 + b_2 \) is reached and 2 symbols in node \( n_2 \) till \( b_0 \) is obtained to repair \( b_0 \). Therefore a total of 5 reads and a penalty of 4 is necessary, reaching the upper-bound of \((r+1)(r+2)/2 - 2\) above. On the other hand, if \( c_{0,0} \) is to be read in Table [II] while node 0 fails, we only need to read \( c_{1,0} \) from node 1 and \( c_{0,0} + c_{1,0} \) from node 2 to recover \( c_{0,0} \), and the penalty is 1.

F. Code Rate

We define the code rate \( R \) of \( C \) to be the ratio of the number of original message symbols over the total number of storage units required to store the encoded symbols, that is

\[
R = \frac{M}{\alpha n} = \frac{M^*}{\alpha} = \frac{((\Delta - 1)(r + \delta - 1) + r\Delta_1 + \delta - 1)\alpha}{(\Delta - 1)r + r\Delta_1 - 1},
\]

where \( R^{(1)} = M^*/((\Delta - 1)r + r\Delta_1 - 1) \) is the code rate of code \( C^{(1)} \). Hence the code rate of \( C \) is bounded by that of \( C^{(1)} \), and the factor \( 1/(\Delta - 1)(r + \delta - 1) + r\Delta_1 - 1 + \Delta_1 - 1 \) reflects the cost of extra storage to obtain the \( (r, \delta) \) repair locality. Note when we set \( \delta = 2 \) and \( (r + 1) | n \), we have \( R = R^{(1)} \), the same as that presented in [7].

IV. Conclusion

In this paper, we propose locally repairable codes that achieve the optimal minimum distance. A two-layer encoding structure composed of a scalar MDS and an MDS array code achieves the desired data reconstruction and repair locality. Any existing MDS codes can be used in the first layer, hence our construction is guaranteed in a finite field with size growing linearly with respect to the total number of storage nodes. The structure also facilitates efficient degraded reads in the DSS.

REFERENCES

[1] H. Weatherspoon and J. D. Kubiatowicz, “Erasure coding vs. replication: A quantitative comparison,” in Proc. Int. Workshop Peer-to-Peer Syst., Cambridge, MA, USA, March 2002.

[2] A. G. Dimakis, P. B. Godfrey, M. J. Wainwright, and K. Ramchandran, “Network coding for distributed storage systems,” in IEEE INFOCOM 2007, Anchorage, AK, 2007.

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

[4] F. Oggier and A. Datta, “Self-repairing homomorphic codes for distributed storage systems,” in IEEE INFOCOM 2011, Shanghai, China, April 2011, pp. 1215–1223.

[5] O. Khan, R. Burns, J. Plank, and C. Huang, “In search of I/O-optimal recovery from disk failures,” in Hot Storage 2011, Portland, OR, Jun. 2011.

[6] P. Gopalan, C. Huang, H. Simitci, and S. Yekhanin, “On the locality of codeword symbols,” IEEE Trans. Info. Theory, vol. 58, no. 11, pp. 6925–6934, Nov. 2012.

[7] D. S. Papailiopoulos and A. G. Dimakis, “Locally repairable codes,” in Proc. IEEE Int. Symp. on Information Theory, Cambridge, MA, July 2012.

[8] I. Tamo, D. S. Papailiopoulos, and A. G. Dimakis, “Optimal locally repairable codes and connection to matroid theory,” February 2013, available online at http://arxiv.org/abs/1301.7693v2.

[9] G. M. Kamath, N. Prakash, V. Lalitha, and P. V. Kumar, “Codes with local regeneration,” in 2013 Information Theory and Applications Workshop (ITA 2013), San Diego, USA, Feb. 2013.

[10] A. S. Rawat, N. Silberstein, O. O. Koyluoglu, and S. Vishwanath, “Optimal locally repairable codes with local minimum storage regeneration via rank-metric codes,” in Information Theory and Applications Workshop (ITA), San Diego, CA, Feb. 2013.

[11] E. M. Gabidulin, “Theory of codes with maximum rank distance,” Problems of Information Transmission, vol. 21, no. 1, pp. 1–12, January 1985.

[12] S. E. Rouayheb and K. Ramchandran, “Fractional repetition codes for repair in distributed storage systems,” in 48th Annual Allerton Conference on Communication, Control, and Computing, Urbana Champaign, IL, Sep. 2010, pp. 1510–1517.

[13] O. Olmez and A. Ramamoorthy, “Replication based storage systems with local repair,” May 2013, available online at http://arxiv.org/abs/1305.5764.

[14] C. Huang, M. Chen, and J. Li, “Pyramid codes: Flexible schemes to trade space for access efficiency in reliable data storage systems,” in Proc. 6th IEEE Int. Symp. Netw. Comput. Appl. (NCA), 2007, pp. 79–86.

[15] A. S. Rawat, O. O. Koyluoglu, N. Silberstein, and S. Vishwanath, “Optimal locally repairable and secure codes for distributed storage systems,” August 2013, available online at http://arxiv.org/abs/1210.6954v2.

[16] I. S. Reed and G. Solomon, “Polynomial codes over certain finite fields,” J. Soc. Indus. Appl. Math., vol. 8, pp. 300–304, 1960.

[17] Y. Wu and A. G. Kimakis, “Reducing repair traffic for erasure coding-based storage via interference alignment,” in Proc. IEEE Int. Symp. on Information Theory, Seoul, Korea, June 2009, pp. 2276–2280.

[18] C. Suh and K. Ramchandran, “Exact-repair MDS codes for distributed storage using interference alignment,” in IEEE Int. Symp. Info. Theory, Austin, Texas, USA, June 2010, pp. 161–165.

[19] 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. Info. Theory, vol. 57, no. 8, pp. 5227–5239, August 2011.

[20] H. Xie and Z. Yan, “MDS codes with low repair complexity for distributed storage networks,” in 2013 22nd Wireless and Optical Communication Conference (WOCC 2013), Chongqing, China, May 16-18 2013.

[21] O. Khan, R. Burns, J. Plank, W. Pierce, and C. Huang, “Rethinking erasure codes for could file system: Minimizing I/O for recovery and degraded reads,” in 10th USENIX Conference on File and Storage Technologies (FAST’12), San Jose, CA, Feb. 2012.