A Novel Construction of Low-Complexity MDS Codes with Optimal Repair Capability for Distributed Storage Systems

Sheng Guan, Haibin Kan, Member, IEEE, and Xin Wang, Senior Member, IEEE

Abstract—Maximum-distance-separable (MDS) codes are a class of erasure codes that are widely adopted to enhance the reliability of distributed storage systems (DSS). In \((n, k)\) MDS coded DSS, the original data are stored into \(n\) distributed nodes in an efficient manner such that each storage node only contains a small amount (i.e., \(1/k\)) of the data and a data collector connected to any \(k\) nodes can retrieve the entire data. On the other hand, a node failure can be repaired (i.e., stored data at the failed node can be successfully recovered) by downloading data segments from other surviving nodes. In this paper, we develop a new approach to construction of simple \((5, 3)\) MDS codes. With judiciously block-designed generator matrices, we show that the proposed MDS codes have a minimum stripe size \(\alpha = 2\) and can be constructed over a small (Galois) finite field \(\mathbb{F}_4\) of only four elements, both facilitating low-complexity computations and implementations for data storage, retrieval and repair. In addition, with the proposed MDS codes, any single node failure can be repaired through interference alignment technique with a minimum bandwidth. The low-complexity and all-node-optimal-repair properties of the proposed MDS codes make them readily deployed for practical DSS.

Index Terms—Distributed storage systems, interference alignment, maximum-distance-separable codes, repair bandwidth.

I. INTRODUCTION

To ensure the reliability of distributed storage systems (DSS), erasure codes can be employed to encode the data segments across the storage nodes. Relying on erasure codes, a data collector can reconstruct the entire data in the presence of (multiple) node failures in DSS. Maximum-distance-separable (MDS) codes are a class of erasure codes. To facilitate low-complexity implementation, systematic MDS codes are widely adopted for DSS. Specifically, if \((n, k)\) systematic MDS codes are used, then the entire raw data of \(M\) symbols are divided into \(k\) segments of \(\alpha = M/k\) symbols. The \(k\) (raw) data segments are directly stored in \(k\) “systematic” nodes, and linear combinations of these \(k\) segments are stored in \(n - k\) “parity” nodes, such that a data collector connected to any \(n\) nodes can reconstruct the entire data. In other words, data retrieval for the MDS coded DSS can tolerate no more than \(n - k\) node failures. Fig.1 presents an example of \((4, 2)\) MDS coded DSS, where it can be readily verified that the entire data can be reconstructed by a data collector connected to every two nodes, and every single node failure can be repaired by a newcomer (i.e., a replacement) downloading data from other three nodes.

Building on MDS codes, Dimakis et al. [1] proposed a new class of regenerating codes to achieve the efficient repair of node failures for DSS. Two special cases of regenerating codes, minimum storage regenerating (MSR) codes and minimum bandwidth regenerating (MBR) codes, were developed. It was also shown that the theoretic lower bound of the repair bandwidth (i.e., the total data amount that is required to be downloaded) with MSR codes for repairing single node failures is given by [1]

\[
\gamma_{\text{min}} = \frac{M}{k} \cdot \frac{n-1}{n-k}.
\]  

Rashimi et al. [2] developed a product-matrix framework to construct MSR codes that allow exact-repair of every single node failure using such a minimum bandwidth. The MSR codes in [2] have a code rate of \(k/n \leq k/(2k-1)\) (whose upper bound is close to 1/2 as \(k\) grows large) and need to be constructed over a finite field of size \(q \geq n(n-k)\). Y. Wu [3] further introduced a construction of functional-repair MDS codes achieving the optimal repair bandwidth, where a part of data is allowed to be changed after repair as long as the ability to retrieve the original data remains. Functional-repair in DSS is usually not preferred in practice since change of data can lead to an increasing data access time. Shah et al. [4] relied on interference alignment techniques to construct \((2k, k)\) exact-repair MDS codes. A disadvantage of their codes is that only the systematic nodes can be optimally repaired whereas repair of a parity node requires downloading data of the original size (i.e., \(M\) symbols). C. Suh and K.
Ramchandran [5] improved the construction in [4] by using the technique of common eigenvectors, so that their codes possess all-node-repair capability using minimum repair bandwidth. S. Liang et al. [6] also provided a construction of all-node-optimal exact-repair (6, 3) MSR codes over the finite field \( \mathbb{F}_2 \). The code rate of the MSR codes in [4], [5], [6] is 1/2, implying that the same amount (\( M \) symbols) of “redundant” data need to be stored to ensure the reliability of DSS.

To build more efficient DSS, [8] and [10] proposed constructions of \((k + 2, k)\) MDS codes with optimal repair bandwidth. Papailiopoulos et al. [8] used Hadamard matrices to enable perfect interference alignment in repair of node failures. Their codes can be constructed over a finite field of size \( q \geq 2k + 3 \), and ensure that all single node failures can be optimally repaired using minimum bandwidth. On the other hand, Tamo et al. [10] employed zigzag codes to ensure exact-repair of systematic nodes optimally, and their zigzag codes can be constructed over a small finite field \( \mathbb{F}_4 \). Codes in [8] and [10] require the stripe size \((i.e., \text{the number of stored data symbols per node}) \alpha = 2^{k+1} \) and \( \alpha = 2^{k-1} \), respectively. It was shown in [11] that a small stripe size facilitates flexible and low-cost operations of DSS; hence, it is preferred in practice. To this end, [9] developed a construction of \((6, 4)\) MDS codes with the minimum stripe size \( \alpha = 2 \), whereas [4] also derived a construction of \((6, 3)\) MDS codes with \( \alpha = 3 \). As with the codes in [10], these two constructions can only ensure optimal repair of systematic nodes.

In this paper, we develop a novel construction of \((5, 3)\) MDS codes with the minimum stripe size \( \alpha = 2 \). With judiciously block-designed generator matrices, we show that the proposed MDS codes can be constructed over a small (Galois) finite field \( \mathbb{F}_q \) of only four elements. In addition, we rigorously establish that the proposed MDS codes can ensure the repair of any single node failure through interference alignment technique using the minimum bandwidth; i.e., the proposed codes are indeed MSR codes. Tab. I compares the proposed MDS codes with the existing alternatives in terms of stripe size, (construction) field size, and node-repair capability. Note that the \((5, 3)\) Hadamard based MDS codes [8] require the stripe size \( \alpha = 16 \) and need to be constructed over a finite field of (at least) size 9, which are much larger than the stripe size \( \alpha = 2 \) and field size \( q = 4 \) with the proposed MDS codes. The \((5, 3)\) zigzag codes \([10], (6, 3)\) MDS codes \([4]\), and \((6, 4)\) MDS codes \([9]\) have stripe sizes \( \alpha = 4, 3, 2 \), respectively; yet, they can only ensure optimal repair of systematic nodes. Compared to the existing codes, the proposed MDS codes have optimal repair capability for both systematic nodes and parity nodes, and facilitate low-complexity computations and implementations for data storage, retrieval and repair. They can be attractive MDS code candidates for practical DSS.

The rest of the paper is organized as follows. In Sec. II, we describe the construction of the proposed MDS codes. Sec. III proves that the proposed MDS codes ensure the optimal exact repairs of all single node failures. Sec. IV provides examples of the proposed MDS codes. Sec. V concludes the paper.

Notations: The following notations are used throughout the paper. Boldface letters denote vectors or matrices; \( T \) denotes transpose; \( \text{rank}(A) \) denotes the rank of a matrix \( A \); \( I \) denotes an identity matrix with a proper size; \( 0 \) denotes an all-zero vector (or matrix) with a proper size; \( \mathbb{F}_q \) denotes a Galois finite field of \( q \) elements, and when it is clear from the context, addition, subtraction, multiplication and division are defined over Galois finite field.

## II. Code Construction

In this section, we delineate the explicit construction of our \((5, 3)\) MDS codes with a stripe size of 2; i.e., here we have \( n = 5, k = 3, \) and \( \alpha = 2 \). The proposed codes are used for DSS with \( n = 5 \) nodes to store a data file containing \( M = k\alpha = 6 \) symbols over a certain finite field \( \mathbb{F}_q \). Each node needs to store a (coded) data segment of \( \alpha = 2 \) symbols. Note that one symbol can be a data block of many (e.g., 4000) bits.

Denote the entire data file by a \( 6 \times 1 \) vector \( m := [m_1^T, m_2^T, m_3^T]^T \), where \( m_i, i = 1, 2, 3, \) are \( 2 \times 1 \) vectors. Similar to [8], [9], we adopt a \( \alpha \times k\alpha \) (i.e., \( 10 \times 8 \)) generator matrix in the form:

\[
P = \begin{bmatrix}
I & I & I \\
A_1^T & A_2^T & A_3^T
\end{bmatrix}
\]

where \( I \) denotes a \( 2 \times 2 \) identity matrix, and \( A_i, i = 1, 2, 3, \) are \( 2 \times 2 \) matrices to be designed. With the generator matrix \( P \), the encoded data can be collected into a \( 10 \times 1 \) vector \( u := [u_1^T, u_2^T, u_3^T, u_4^T, u_5^T]^T \), where \( u_i, i = 1, \ldots, 5 \), are \( 2 \times 1 \) vectors. Then we have

\[
u = P \cdot m = \begin{bmatrix}
m_1 \\
m_2 \\
m_3 \\
A_1^T m_1 + A_2^T m_2 + A_3^T m_3
\end{bmatrix} = \begin{bmatrix}
u_1 \\
u_2 \\
u_3 \\
u_4 \\
u_5
\end{bmatrix}.
\]

The symbols in \( u_i \) are exactly the data stored in node \( i = 1, \ldots, 5 \). Clearly, data stored in the first \( k = 3 \) nodes are in raw form; hence, these nodes are called systematic nodes. On the other hand, the rest 2 nodes are called parity nodes as their data are linear combinations of the raw data segments.

### TABLE I

| Codes | (5, 3) Hadamard MDS codes [8] | (5, 3) Zigzag MDS codes [10] | (5, 3) MSR codes [21] | (6, 3) MDS codes [4] | (6, 4) MDS codes [9] | Proposed (5, 3) MDS codes |
|-------|--------------------------------|-------------------------------|----------------------|----------------------|----------------------|-------------------------|
| Stripe size | 16                              | 4                             | 3                    | 2                    | 2                    | Yes                     |
| Field size   | \( > 9 \)                       | \( > 3 \)                     | \( > 10 \)           | \( > 6 \)            | \( > 4 \)            | \( > 4 \)               |
| Optimal repair | Yes                             | No                            | Yes                  | No                   | No                   | Yes                     |
Construction of the proposed MDS codes boils down to designing proper matrices \( A_1, A_2, \) and \( A_3 \) such that a data collector connected to any \( k = 3 \) nodes can retrieve the entire original data file, and any single node failure can be repaired using minimum bandwidth.

### A. Construction of \( A_1, A_2, \) and \( A_3 \) over \( \mathbb{F}_4 \)

Denote the 2-dimensional standard basis vectors by \( e_1 := [1 \ 0]^T \) and \( e_2 := [0 \ 1]^T \). Select four parameters \( \lambda, \mu, \theta, \eta \) over a certain finite field \( \mathbb{F}_q \), and let

\[
\begin{align*}
A_1 &= \begin{bmatrix}
\theta e_1^T + \eta e_2^T \\
\eta e_1^T + \mu e_2^T
\end{bmatrix} = \begin{bmatrix}
\theta & 0 \\
\eta & 1
\end{bmatrix}, \\
A_2 &= \frac{1}{\lambda} \begin{bmatrix}
(\theta - 1)e_1^T \\
\eta e_1^T + e_2^T
\end{bmatrix} = \frac{1}{\lambda} \begin{bmatrix}
\theta - 1 & 0 \\
\eta & 1
\end{bmatrix}, \\
A_3 &= \frac{1}{\mu} \begin{bmatrix}
\theta e_1^T \\
\eta e_1^T + \mu e_2^T
\end{bmatrix} = \frac{1}{\mu} \begin{bmatrix}
\theta & -1 \\
\eta & 1
\end{bmatrix}.
\end{align*}
\]  

(4)

Note that there exists a relationship among \( A_1^T, A_2^T \) and \( A_3^T \), which is

\[
A_1^T = \lambda A_2^T + [e_1 \ 0] = \mu A_3^T + [e_2 \ 0],
\]  

(5)

where 0 denotes a zero vector of length 2. Such a construction can facilitate the implementation of interference alignment to perform optimal node repair using minimum bandwidth, as will be shown in the sequel.

In order to obtain the desired MDS codes, we require that the parameters \( \{\lambda, \mu, \theta, \eta\} \) satisfy the following conditions:

\[
\begin{align*}
\lambda &\neq 0, 1; \\
\mu &\neq 0, 1; \\
\theta &\neq 0, 1; \\
\eta &\neq 0, -1; \\
\theta &\neq 0; \\
\theta(1 - \lambda) &\neq 1; \\
\theta(\mu - 1) &\neq \eta; \\
\theta(\mu - 1) &\neq \eta \lambda + \mu; \\
(\mu - 1) &\neq \eta(1 - \lambda); \\
\eta + 1 &\neq \frac{(\theta - 1)(\mu - \lambda)}{\theta(\mu - 1)}. \\
\end{align*}
\]  

(6a)–(6j)

Interestingly, we can show that the matrices \( A_1, A_2, \) and \( A_3 \) in (4) under conditions (6a)–(6j) can be actually constructed over a finite field as small as \( \mathbb{F}_4 \). To see it, we use \( \mathbb{F}_4 \) generated by primitive polynomial \( w^2 + w + 1 \) for illustration. We denote elements in this \( \mathbb{F}_4 \) by \( \{0, 1, w, w + 1\} \). Note that the opposite of each element corresponds to itself; i.e., \(-1 \in \mathbb{F}_4 \) equals to \( 1 \).

### 1. Tab. II provides the product table for these four elements. We can establish that

**Theorem II.1.** The proposed codes can be constructed over finite field \( \mathbb{F}_4 \).

**Proof.** In \( \mathbb{F}_4 \) generated by primitive polynomial \( w^2 + w + 1 \), let \( \eta = \lambda = w \) and \( \theta = \mu = w + 1 \). Then the conditions (6a)–(6d) clearly hold. Furthermore, we have

\[
\begin{align*}
\theta + \eta &\neq 1, \\
\theta(1 - \lambda) &\neq (w + 1)(1 + 1) = w \neq 1, \\
\theta(\mu - 1) &\neq (w + 1)w = 1 \neq \eta, \\
\theta(\mu - \lambda) &\neq (w + 1)w \neq \eta \lambda + \mu = w \cdot w + 1 = 0, \\
(\mu - 1) &\neq w \neq \eta(1 - \lambda) = w(1 + w) = 1, \\
\eta + 1 &\neq w + 1 \neq \frac{(\theta - 1)(\mu - \lambda)}{\theta(\mu - 1)} = w.
\end{align*}
\]

In other word, conditions (6e)–(6i) are satisfied. It is also easy to see that with \( \eta = \lambda = w \) and \( \theta = \mu = w + 1 \), all entries of \( A_1, A_2, \) and \( A_3 \) in (4) are either 0, 1, \( w \), or \( w + 1 \).

### B. MDS Property

We next show that the proposed codes are in fact MDS codes. To this end, we first establish the following two lemmas for the proposed matrices \( A_1, A_2, \) and \( A_3 \).

**Lemma II.2.** \( A_1, A_2, \) and \( A_3 \) have full rank.

**Proof.** By conditions (6c)–(6e) and \( \lambda, \mu \neq 0 \) in (4), we can readily derive rank(\( A_1 \)) = rank(\( A_2 \)) = rank(\( A_3 \)) = 2.

**Lemma II.3.** \( A_1 - A_2, A_1 - A_3, \) and \( A_2 - A_3 \) have full rank.

**Proof.** Since \( \lambda, \mu \neq 0 \), we have

\[
\begin{align*}
\text{rank}(A_1 - A_2) &= \text{rank}(\frac{1}{\lambda} \begin{bmatrix}
\theta \lambda - \theta + 1 \\
\eta(\lambda - 1)
\end{bmatrix}) \\
&= \text{rank}(\begin{bmatrix}
\theta \lambda - \theta + 1 \\
\eta(\lambda - 1)
\end{bmatrix}) = 2.
\end{align*}
\]

due to \( \lambda \neq 1 \) and \( 1 - \theta(1 - \lambda) = \theta \lambda - \theta + 1 \neq 0 \) by (6f);

\[
\begin{align*}
\text{rank}(A_1 - A_3) &= \text{rank}(\frac{1}{\mu} \begin{bmatrix}
\theta \mu - 1 \\
\eta(\mu - 1)
\end{bmatrix}) \\
&= \text{rank}(\begin{bmatrix}
\theta \mu - 1 \\
\eta(\mu - 1)
\end{bmatrix}) = 2.
\end{align*}
\]

due to \( \mu \neq 1 \) and \( \theta(\mu - 1) - \eta \neq 0 \) by (6g);

\[
\begin{align*}
\text{rank}(A_2 - A_3) &= \text{rank}(\begin{bmatrix}
\frac{\theta - 1}{\lambda} - \frac{\theta}{\mu} \\
\frac{\eta}{\lambda} - \frac{\eta}{\mu}
\end{bmatrix}) \\
&= \text{rank}(\begin{bmatrix}
\frac{\theta - 1}{\lambda} - \frac{\theta}{\mu} \\
\frac{\eta}{\lambda} - \frac{\eta}{\mu}
\end{bmatrix}) = 2.
\end{align*}
\]

due to \( \lambda \neq \mu \) and \( \theta(\mu - \lambda) - \eta \lambda - \mu = \lambda \mu(\frac{\theta - 1}{\lambda} - \frac{\theta + 2}{\mu}) \neq 0 \) by (6h).

Based on Lemmas II.2 and II.3, we can then show

**Table II**

The product table over \( \mathbb{F}_4 \) generated by primitive polynomial \( w^2 + w + 1 \)

|   | 0  | 1  | w  | w + 1 |
|---|----|----|----|-------|
| 0 | 0  | 1  | w  | w + 1 |
| 1 | 1  | 0  | w  | w + 1 |
| w | 0  | w  | w + 1 | 1     |
| w + 1 | 0 | w + 1 | 1 | w     |
Theorem II.4. The proposed codes have MDS property.

Proof. We prove the lemma by checking whether the proposed codes can ensure that the entire data can be retrieved by using data segments at arbitrary three nodes. With the proposed codes, if a data collector connects to the three systematic nodes or any two systematic nodes and the first parity node, then clearly the entire data can be readily retrieved. Suppose that the data collector connects to the first and the second systematic nodes, as well as the second parity node. Then it can obtain

\[
\begin{bmatrix}
u_1 \\ u_2 \\ u_3 \\ u_5 \\
\end{bmatrix} = \begin{bmatrix}
I & I \\
A_1^T & A_2^T & A_3^T \\
\end{bmatrix} m.
\]  

(8)

It follows from Lemmas II.2 that \( A_3 \) has full rank; hence, the big matrix before \( m \) in (8) is invertible, and the original data \( m \) can be retrieved. Similarly, as \( A_2 \) (or \( A_1 \)) has full rank by Lemmas II.2, the original data \( m \) can be retrieved when the data collector connects to the first (or second) and the third systematic nodes, as well as the second parity node. Furthermore, as \( A_1 - A_2, A_1 - A_3 \) and \( A_2 - A_3 \) have full rank by Lemmas II.3, we can show that the original data \( m \) can be retrieved when the data collector connects to one of the systematic nodes and two parity nodes. As the entire data can be retrieved by the data collector connected to arbitrary three nodes, the proposed codes are MDS codes.

Theorems II.1 and II.4 establish that the proposed codes are MDS codes that can be constructed over a small finite field \( \mathbb{F}_4 \). With such a construction, the computational complexity of retrieving the original data file is low. It is clear that data retrieval with the three systematic nodes consumes no computation. For retrieval with two systematic nodes and one parity node, take (8) as an example. In this case, we have

\[
m = \begin{bmatrix}
I & I \\
A_1^T & A_2^T & A_3^T \\
\end{bmatrix}^{-1} \begin{bmatrix}
u_1 \\ u_2 \\ u_3 \\ u_5 \\
\end{bmatrix} = \begin{bmatrix}
I \\
-A_3^{-T} A_1^T & -A_3^{-T} A_2^T & A_3^{-T} \\
\end{bmatrix} \begin{bmatrix}
u_1 \\ u_2 \\ u_3 \\ u_5 \\
\end{bmatrix} = \begin{bmatrix}
u_1 \\ -A_3^{-T} A_1^T u_1 - A_3^{-T} A_2^T u_2 + A_3^{-T} u_5 \\
\end{bmatrix}.
\]

Recovering \( m \) only requires calculating \(-A_3^{-T} A_1^T u_1 - A_3^{-T} A_2^T u_2 + A_3^{-T} u_5\). Note that the coefficient matrices \(-A_3^{-T} A_1^T, A_3^{-T} A_2^T, \) and \( A_3^{-T} \) can be pre-computed offline.\(^1\)

Hence, the computation for online retrieval consists of \( 2k \) multiplications and \((2k - 1)\alpha \) additions. With \( M = k\alpha \), the overall complexity is \( O(M\alpha) \). For the worst case where two parity nodes are involved to reconstruct the original file, it was shown that the data retrieval can be still performed with a computational complexity of \( O(M\alpha) \). With \( \alpha = 2 \) in the proposed codes, such a complexity is clearly low. In addition,

\(^1\)Even if they need to be computed online, the computational complexity is low as \( A_1, A_2, \) and \( A_3 \) are all \( 2 \times 2 \) matrices.

III. NODE REPAIR THROUGH INTERFERENCE ALIGNMENT

In this paper, we mainly concentrate on the repair of a single node failure since such a case occurs at most of time in practice. In this section, we rigorously establish that the proposed MDS codes have optimal repair capability; i.e., they are MSR codes that can ensure exact repair of every single node failure using minimum bandwidth. From (1), we can derive the minimum repair bandwidth in our scenario as

\[
\gamma_{min} = \frac{M}{k} \cdot \frac{n - k}{n - k} = \alpha \cdot \frac{k + 1}{2} = 4 \text{ (symbols).} \quad (9)
\]

To achieve this optimal repair bandwidth, a key technique that we employ for node repair is interference alignment.

A. Repair of Systematic Nodes

The interference alignment technique was first proposed to approach the capacity of interference channels in the context of wireless communications [12], [13], and it was later introduced into MDS code design for DSS by Y. Wu et al. [7]. We first show that optimal repair of systematic nodes can be achieved with the proposed MDS codes through interference alignment.

Assume that the \( i \)th \((i = 1, 2, 3)\) systematic node fails. By (9), repair of a node failure is allowed to download only one symbol per node from the four surviving nodes. Let \( \varphi_{i1}^T \) and \( \varphi_{i2}^T \) denote two \( 1 \times 2 \) vectors. From the first and second parity nodes, the newcomer can download one symbol \( d_{i1} \) and one symbol \( d_{i2} \), respectively, as follows.

\[
\begin{bmatrix} d_{i1} \\ d_{i2} \end{bmatrix} = \begin{bmatrix} \varphi_{i1}^T u_{k+1} \\ \varphi_{i2}^T u_{k+2} \end{bmatrix} = \begin{bmatrix} \varphi_{i1}^T A_{i1}^T \\ \varphi_{i2}^T A_{i2}^T \end{bmatrix} m_i + \sum_{j \neq i} \begin{bmatrix} \varphi_{i1}^T A_{i1}^T \\ \varphi_{i2}^T A_{i2}^T \end{bmatrix} m_j. \quad (10)
\]

Note that symbols \( d_{i1} \) and \( d_{i2} \) are linear combinations of symbols stored in the first and the second parity nodes. Clearly, the first term containing \( m_i \) in (10) is the desired “signal” and the other term is “interference” which should be aligned. As the surviving two systematic nodes \( j \) contain \( m_j, j \neq i \), the newcomer can further download data from these two nodes to cancel out the “interference” in (10). However, to achieve minimum repair bandwidth, the newcomer can only download
one symbol per node. To perform optimal repair through interference alignment, the following conditions must hold:

\[
\text{rank}\left(\begin{bmatrix} \varphi_{T1}^T \\ \varphi_{T2} A_{T1}^T \end{bmatrix}\right) = 2 \\
\text{rank}\left(\begin{bmatrix} \varphi_{T4}^T \\ \varphi_{T2} A_{T1}^T \end{bmatrix}\right) = 1, \quad j \neq i
\] (11)

As the newcomer wants to recover \(m_i\) stored in failed nodes, the matrix before \(m_i\) in (10) should have full rank so that \(m_i\) might be repaired by pre-multiplying the inverse of this matrix after interference cancellation. This also implies that \(\varphi_{T1}^T\) should have full row rank. The other conditions then actually require that the row spaces of \(\varphi_{T1}^T\) and \(\varphi_{T2} A_{T1}^T\) to fully overlap. Under these conditions, the newcomer can download only one linear combination of symbols at each systematic node, which can be either \(\varphi_{T1}^T m_j\) or \(\varphi_{T2} A_{T1}^T m_j\), to (align with and) cancel the interference.

Now in order to show that the systematic nodes can be optimally repaired, we only need to prove that there exist \(\varphi_{T1}^T\) and \(\varphi_{T2}^T\), \(\forall i = 1, 2, 3\), such that the conditions in (11) hold. This is established in the following three lemmas.

**Lemma III.1.** For \(i = 3\), the conditions in (11) hold with \(\varphi_{T2}^T = e_{T2}^T\) and \(\varphi_{T1}^T = \varphi_{T2} A_{T1}^T\).

**Proof.** Clearly \(e_{T2}^T \varphi_{T3} = 0\) and \(e_{T2}^T \varphi_{T3} = 1\). Then we have

\[
A_1 \varphi_{T3} = \begin{bmatrix} \theta e_{T1}^T \\ \eta e_{T1}^T + e_{T2}^T \end{bmatrix}, \quad \varphi_{T3} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}
\]
\[
A_2 \varphi_{T3} = \frac{1}{\lambda} \begin{bmatrix} (\theta - 1) e_{T1}^T \\ \eta e_{T1}^T + e_{T2}^T \end{bmatrix}, \quad \varphi_{T3} = \begin{bmatrix} 0 \\ -1 \end{bmatrix}
\]
\[
A_3 \varphi_{T3} = \frac{1}{\mu} \begin{bmatrix} \theta e_{T1}^T - e_{T2}^T \\ \eta e_{T1}^T + e_{T2}^T \end{bmatrix}, \quad \varphi_{T3} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}
\] (12)

Since \(\varphi_{T1}^T = \varphi_{T2}^T A_{T1}^T = e_{T1}^T A_{T1}^T\), and \(\lambda, \mu \neq 1\) in (4), we further have

\[
\text{rank}\left(\begin{bmatrix} \varphi_{T1}^T \\ \varphi_{T2} A_{T1}^T \end{bmatrix}\right) = \text{rank}\left(\begin{bmatrix} 0 \\ 1 \end{bmatrix}\right) = 1
\]
\[
\text{rank}\left(\begin{bmatrix} \varphi_{T1}^T \\ \varphi_{T2} A_{T1}^T \end{bmatrix}\right) = \text{rank}\left(\begin{bmatrix} 0 \\ 1/\lambda \end{bmatrix}\right) = 1
\] (13)
\[
\text{rank}\left(\begin{bmatrix} \varphi_{T1}^T \\ \varphi_{T2} A_{T1}^T \end{bmatrix}\right) = \text{rank}\left(\begin{bmatrix} -1/\mu \\ 1/\mu \end{bmatrix}\right) = 2
\]

The proof completes.

**Lemma III.2.** For \(i = 2\), the conditions in (11) hold with \(\varphi_{T2}^T = e_{T2}^T\) and \(\varphi_{T1}^T = \varphi_{T2} A_{T1}^T\).

**Proof.** As \(e_{T1}^T \varphi_{T2} = 1\) and \(e_{T2}^T \varphi_{T2} = 0\), we have

\[
A_1 \varphi_{T2} = \begin{bmatrix} \theta e_{T1}^T \\ \eta e_{T1}^T + e_{T2}^T \end{bmatrix}, \quad \varphi_{T2} = \begin{bmatrix} \theta \\ \eta \end{bmatrix}
\]
\[
A_2 \varphi_{T2} = \frac{1}{\lambda} \begin{bmatrix} (\theta - 1) e_{T1}^T \\ \eta e_{T1}^T + e_{T2}^T \end{bmatrix}, \quad \varphi_{T2} = \frac{1}{\lambda} \begin{bmatrix} \theta \\ \eta \end{bmatrix}
\] (14)

Since \(\varphi_{T2}^T = \varphi_{T2} A_{T1}^T = e_{T1}^T A_{T1}^T\), \(\lambda, \mu \neq 0\) in (4), and \(\theta, \eta \neq 0\) in (6c)–(6d), we can derive

\[
\text{rank}\left(\begin{bmatrix} \varphi_{T1}^T \\ \varphi_{T2} A_{T1}^T \end{bmatrix}\right) = \text{rank}\left(\begin{bmatrix} \theta \\ \eta \end{bmatrix}\right) = 1
\]
\[
\text{rank}\left(\begin{bmatrix} \varphi_{T2}^T \\ \varphi_{T2} A_{T1}^T \end{bmatrix}\right) = \text{rank}\left(\begin{bmatrix} \theta/\mu \\ \eta/\mu \end{bmatrix}\right) = 1
\] (15)

The lemma follows.

**Lemma III.3.** For \(i = 1\), the conditions in (11) hold with \(\varphi_{T1}^T = e_{T1}^T + e_{T2}^T\) and \(\varphi_{T1} = \varphi_{T2} A_{T1}^T\).

**Proof.** It is clear that \(e_{T1}^T \varphi_{T2} = e_{T2}^T \varphi_{T2} = 1\). Then we have

\[
A_1 \varphi_{T1} = \begin{bmatrix} \theta e_{T1}^T \\ \eta e_{T1}^T + e_{T2}^T \end{bmatrix}, \quad \varphi_{T1} = \begin{bmatrix} \theta + 1 \\ \eta + 1 \end{bmatrix}
\]
\[
A_2 \varphi_{T1} = \frac{1}{\lambda} \begin{bmatrix} (\theta - 1) e_{T1}^T \\ \eta e_{T1}^T + e_{T2}^T \end{bmatrix}, \quad \varphi_{T1} = \frac{1}{\lambda} \begin{bmatrix} \theta - 1 \\ \eta + 1 \end{bmatrix}
\] (16)
\[
A_3 \varphi_{T1} = \frac{1}{\mu} \begin{bmatrix} \theta e_{T1}^T - e_{T2}^T \\ \eta e_{T1}^T + e_{T2}^T \end{bmatrix}, \quad \varphi_{T1} = \frac{1}{\mu} \begin{bmatrix} \theta - 1 \\ \eta + 1 \end{bmatrix}
\]

Since \(\lambda, \mu \neq 0, \theta \neq 0\) in (6c) and \(\eta \neq -1\) in (6d), we have

\[
\text{rank}\left(\begin{bmatrix} \varphi_{T1}^T \\ \varphi_{T2} A_{T1}^T \end{bmatrix}\right) = \text{rank}\left(\begin{bmatrix} \theta/\lambda \\ \eta/\lambda \end{bmatrix}\right) = 1
\]
\[
\text{rank}\left(\begin{bmatrix} \varphi_{T1}^T \\ \varphi_{T2} A_{T1}^T \end{bmatrix}\right) = \text{rank}\left(\begin{bmatrix} \theta/\lambda \\ \eta/\lambda \end{bmatrix}\right) = 1
\] (17)

Based on Lemmas III.1–III.3, we readily arrive at

**Theorem III.4.** Any systematic node can be repaired with minimum repair bandwidth.

Use the proposed MDS codes over the finite field \(\mathbb{F}_4\) in Tab. II as an example. Fig. 2 illustrates the optimal repair of the
cannot be used directly for the repair of a parity node. Hence, we perform a change of basis as follows. Let

\[ \varphi_{31} = e_3^T A_1^T = [0 \ 1] \]

(019)

to obtain a linear combination of two symbols at the first parity node:

\[ d_{31} = \varphi_{31}^T u_4 = [0 \ 1] m_1 + [0 \ 1] m_2 + [0 \ 1] m_3. \]

Similarly, the newcomer uses coefficient vector \( \varphi_{32} = [0 \ 1] \) to obtain a linear combination of two symbols at the second parity node:

\[ d_{32} = \varphi_{32}^T u_5 = [0 \ 1] m_1 + [0 \ ★] m_2 + [★ ▲] m_3. \]

Then the newcomer downloads one symbol \( f_{31} := [0 \ 1] m_1 \) and \( f_{32} := [0 \ 1] m_2 \) from the first and two systematic nodes, respectively. With the downloaded symbols, the newcomer can cancel out the interference and obtain

\[
\begin{bmatrix}
  d_{31} - f_{31} - f_{32} \\
  d_{32} - f_{31} - ★ \cdot f_{32}
\end{bmatrix} = \begin{bmatrix} 0 & 1 \end{bmatrix} m_3.
\]

Since the matrix before \( m_3 \) has full-rank, the desired symbols (i.e. symbols in \( m_3 \)) can be retrieved by pre-multiplying the inverse of this matrix.

Fig. 3 and Fig. 4 illustrate the optimal repair of the second and the first systematic node, respectively. Similar procedures can be performed to repair these two nodes using minimum bandwidth.

**B. Repair of the First Parity Node**

Different from systematic node case, interference alignment cannot be used directly for the repair of a parity node. Hence, we perform a change of basis as follows. Let

\[
P = \begin{bmatrix}
  1 & 0 & 0 & 1 \\
  1 & 0 & 0 & 1 \\
  1 & 0 & 1 & 0 \\
  1 & 0 & 1 & 0 \\
  ★ ▲ & 1 & 1 & ★ \\
  0 & 1 & 0 & ★ ▲ ▲ ▲
\end{bmatrix}
\]

With \( m := [m_1^T, \ m_2^T, \ m_3^T]^T \), when the third systematic node fails, the newcomer needs to recover the symbols stored in it (i.e., symbols in \( m_3 \)). To this end, the newcomer downloads one symbol from each surviving node. Specifically, it uses coefficient vector \( \varphi_{31} = e_3^T A_1^T = [0 \ 1] \) to obtain a linear combination of two symbols at the first parity node:

\[ d_{31} = \varphi_{31}^T u_4 = [0 \ 1] m_1 + [0 \ 1] m_2 + [0 \ 1] m_3. \]

Now the first parity node looks like a systematic node and the third systematic node looks like a parity node. To optimally repair the first parity node, the newcomer needs to download one symbol from each surviving node. Define \( \varphi_{41}^T \) and \( \varphi_{42}^T \), which are \( 1 \times 2 \) row vectors used to access symbols from the third systematic node and the second parity node, respectively. The newcomer then obtains

\[
\begin{bmatrix}
  0 & 1 & 1 & 1 & 1 & 0 & 1 & 0
\end{bmatrix}
\]

\[ m' = \begin{bmatrix} I & I & I \end{bmatrix} m. \]

As with (11), the following conditions must be satisfied to perform interference alignment using the minimum repair bandwidth.

\[ \text{rank}\left( \begin{bmatrix} \varphi_{41}^T \\ \varphi_{42}^T \end{bmatrix} \right) = 1 \]

\[ \text{rank}\left( \begin{bmatrix} \varphi_{41}^T \\ \varphi_{42}^T \end{bmatrix} \right) = 1 \]

\[ \text{rank}\left( \begin{bmatrix} \varphi_{41}^T \\ \varphi_{42}^T \end{bmatrix} \right) = 2 \]

We prove the following lemma.

**Lemma III.5.** The conditions in (20) hold with \( \varphi_{42} = (\mu - 1) e_2^T + (\lambda - 1) e_1^T \) and \( \varphi_{41}^T = \varphi_{42}^T (A_1^T - A_3^T) \).

**Proof.** It is easy to show:

\[ A_1^T \varphi_{42} = \begin{bmatrix} \theta e_1^T \\ \eta e_1^T + e_2^T \\ \theta(\lambda - 1) \end{bmatrix} \varphi_{42} = \begin{bmatrix} \eta(\lambda - 1) + \mu - 1 \end{bmatrix}, \]

\[ A_2^T \varphi_{42} = \begin{bmatrix} \theta e_1^T \\ \eta e_1^T + e_2^T \\ (\theta - 1)(\lambda - 1) \end{bmatrix} \varphi_{42} = \begin{bmatrix} \eta(\lambda - 1) + \mu - 1 \end{bmatrix}, \]

\[ A_3^T \varphi_{42} = \begin{bmatrix} \theta e_1^T \\ \eta e_1^T + e_2^T \\ \theta(\lambda - 1) - (\mu - 1) \end{bmatrix} \varphi_{42} = \begin{bmatrix} \eta(\lambda - 1) + \mu - 1 \end{bmatrix}. \]
Recall that $A_1^T = \lambda A_2^T + [e_1 \ 0] = \mu A_3^T + [e_2 \ 0]$. Hence, we can further derive

$$(A_1 - A_3)\varphi_{42} = \frac{\lambda(\mu - 1)}{\mu - \lambda} (A_2 - A_3)\varphi_{42} + \left[\frac{\mu - 1}{\mu - \lambda} e_1^T + \frac{\lambda - \mu}{\mu - \lambda} e_2^T\right] \varphi_{42}$$

$$(22) = \frac{\lambda(\mu - 1)}{\mu - \lambda} (A_2 - A_3)\varphi_{42}.$$ 

In addition, we can use (21) to get

$$(A_1 - A_3)\varphi_{42} = \frac{\mu - 1}{\mu} \left[\begin{array}{c} \theta(\lambda - 1) + 1 \\ \eta(\lambda - 1) + \mu - 1 \end{array}\right].$$

As $\mu \neq 0,1$ and $(\mu - 1) \neq \eta(\lambda - 1)$ by (6), i.e., $\eta(\lambda - 1) + (\mu - 1) \neq 0$, we have $\text{rank}(\varphi_{42}^T(A_1^T - A_3^T)) = 1$. Hence,

$\text{rank}(\varphi_{42}^T(A_1^T - A_3^T)) = \text{rank}(\left[\varphi_{42}^T\varphi_{42}^T, e_2^T \varphi_{42}\right]) = 1,$

$\text{rank}(\varphi_{42}^T(A_1^T - A_3^T)) = \text{rank}(\left[\varphi_{42}^T, e_2^T \varphi_{42}\right]) = 1.$

(24)

Since $A_1 = \mu A_3 + [e_2 \ 0]^T (5)$, we have

$\text{rank}(\left[\varphi_{42}^T, e_2^T \varphi_{42}\right]) = \text{rank}(\left[(A_1 - A_3)\varphi_{42}, e_2^T \varphi_{42}\right]) = \text{rank}(\left[\varphi_{42}^T, e_2^T \varphi_{42}\right]) = 1.$

(25)

The last step holds since $\eta(\lambda - 1) + (\mu - 1) \neq 0$ and $\mu \neq 0,1$.

**Theorem III.6.** The first parity node can be optimally repaired using minimum bandwidth.

*Proof.* By Lemma III.5, we have found $\varphi_{41}^T$ and $\varphi_{42}^T$ satisfying (20). Hence, through downloading one symbol from each surviving node and performing interference alignment, the first parity node can be optimally repaired. \(\square\)

Again, construct the proposed MDS codes over the finite field $\mathbb{F}_4$ in Tab. II. Fig. 5 illustrates the optimal repair of the first parity node with the same notations as in Fig. 2. Now the newcomer needs to recover the symbols stored in the first parity node, by downloading one symbol from each surviving node. It uses coefficient vector $\varphi_{41}^T = [\blacklozenge \ \blacklozenge]$ to obtain a linear combination of two symbols at the third systematic node:

$$d_{41} = \varphi_{41}^T u_4 = [\blacklozenge \ \blacklozenge] m_3;$$

and uses coefficient vector $\varphi_{42}^T = [\bigstar \ \blacklozenge]$ to obtain a linear combination of two symbols at the second parity node:

$$d_{42} = \varphi_{42}^T u_5 = [\bigstar \ \blacklozenge] m_1 + [\bigstar \ \blacklozenge] m_2 + [0 \ 1] m_3.$$

Then the newcomer downloads one symbol $f_{s1} : = [1 \ 1] m_1$ and $f_{s2} : = [1 \ 1] m_2$ from the first and two systematic nodes, respectively. With the downloaded symbols, the newcomer can align the interference and obtain

$$\begin{bmatrix} d_{41} + \blacklozenge \cdot f_{s1} + \blacklozenge \cdot f_{s2} \\ d_{42} + \blacklozenge \cdot f_{s1} + \bigstar \cdot f_{s2} \end{bmatrix} = \begin{bmatrix} \blacklozenge \blacklozenge \\ 0 \ 1 \end{bmatrix} \sum_{i=1,2,3} m_i.$$ 

C. Repair of the Second Parity Node

As with repairing the first parity node, a change of basis is needed. Since we have proven the invertibility of $A_1^T$, $A_2^T$ and $A_3^T$ in Lemma II.2, we have

$$u = P \cdot m = \begin{bmatrix} I & I & I \\ A_1^T & A_2^T & A_3^T \end{bmatrix} m$$

$$= \begin{bmatrix} I \\ - (A_3^T)^{-1} A_3^T \\ I \end{bmatrix} m'',$

$$m'' = \begin{bmatrix} I & I & I \\ A_1^T & A_2^T & A_3^T \end{bmatrix} m.$$ 

Now the second parity node looks like a systematic node. To optimally repair the second parity node, we need to access one symbol from each surviving node and interference alignment would be performed to repair the failed node.

Let $\varphi_{51}^T = [0 \ \varphi_{51}^T A_3^T]$ and $\varphi_{52}^T = [0 \ \varphi_{52}^T A_3^T]$, where $\varphi_{51}^T$, $\varphi_{51}^T$, $\varphi_{52}^T$ and $\varphi_{52}^T$ are all $1 \times 2$ row vectors. Using $\varphi_{51}^T$ and $\varphi_{52}^T$, the newcomer obtains two symbols from the third systematic node and the first parity node as follows:

$$\varphi_{51}^T \left[ - (A_3^T)^{-1} A_3^T \right] m'' = \varphi_{51}^T \left[ - A_1^T - A_2^T I \right] m'';$$

$$\varphi_{52}^T \left[ I - (A_3^T)^{-1} A_3^T I - (A_3^T)^{-1} A_2^T \right] m'' = \varphi_{52}^T \left[ A_3^T - A_2^T \right] m''.$$ 

(27)
The following equations should be satisfied in order to perform optimal repair of the second parity node [cf. (11)]:

\[
\begin{align*}
\text{rank}(\begin{bmatrix}
\varphi_{51}^T A_1^T \\
\varphi_{52}^T (A_1^T - A_2^T) \\
\end{bmatrix}) &= 1 \\
\text{rank}(\begin{bmatrix}
\varphi_{51}^T A_2^T \\
\varphi_{52}^T (A_2^T - A_2^T) \\
\end{bmatrix}) &= 1 \\
\text{rank}(\begin{bmatrix}
\varphi_{51}^T \\
\varphi_{52}^T \\
\end{bmatrix}) &= 2
\end{align*}
\]  

(28)

To satisfy the last equality in (28), \(\varphi_{51}^T\) and \(\varphi_{52}^T\) must be non-zero vectors. By Lemmas II.2 and II.3, we have proven the invertibility of \(A_i^T\) and \(A_i^T - A_j^T\), \((i \neq j)\). Hence, \(\varphi_{51}^T A_1^T\), \(\varphi_{51}^T A_2^T\), \(\varphi_{52}^T (A_1^T - A_2^T)\) and \(\varphi_{52}^T (A_3^T - A_2^T)\) are all non-zero vectors. In order to satisfy the first two equalities in (28), we need

\[
\begin{align*}
A_1 \varphi_{51} &= c(A_3 - A_1) \varphi_{52} \\
A_2 \varphi_{52} &= \bar{c}(A_3 - A_2) \varphi_{52},
\end{align*}
\]

where \(c, \bar{c} \neq 0\). That is,

\[
\begin{align*}
A_1^{-1}(A_3 - A_1) \varphi_{52} &= p A_2^{-1}(A_3 - A_2) \varphi_{52},
\end{align*}
\]  

(30)

where \(p = \bar{c}/c\). Rewrite the equation as

\[
((A_1^{-1} A_3 - I) - p (A_2^{-1} A_3 - I)) \varphi_{52} = 0,
\]

(31)

where \(0\) denotes \(2 \times 2\) zero matrix. To obtain a non-zero solution \(\varphi_{52}\) for (31), we need

\[
\text{rank}((A_1^{-1} A_3 - I) - p (A_2^{-1} A_3 - I)) = 1.
\]

(32)

Following the aforementioned lines, we can then find \(\varphi_{51}\) and \(\varphi_{52}\) satisfying (28) as follows:

1) Find \(p \neq 0\) satisfying (32).
2) Obtain a non-zero solution \(\varphi_{52}\) for (31) such that (30) holds.
3) Let \(\varphi_{51} = A_1^{-1}(A_3 - A_1) \varphi_{52}\); then (29) as well as the first two equalities in (28) hold.
4) Check if the last equality in (28) holds.

The following three lemmas show that the aforementioned procedure can successfully obtain the desired \(\varphi_{51}\) and \(\varphi_{52}\).

**Lemma III.7.** Condition (32) holds with \(p = \frac{\mu - 1}{\mu - \lambda}\).

**Proof.** First, we have

\[
\begin{align*}
\text{rank}((A_1^{-1} A_3 - I) - p (A_2^{-1} A_3 - I)) &= \text{rank}(A_1 \cdot ((A_1^{-1} A_3 - I) - p (A_2^{-1} A_3 - I))) \\
&= \text{rank}((p - 1) A_1 + A_3 - p A_1 A_2^{-1} A_3)
\end{align*}
\]

(33)

By (4),

\[
A_2 = \frac{1}{\lambda} \begin{bmatrix}
(\theta - 1) e_1^T \\
\eta e_1^T + e_2^T
\end{bmatrix}.
\]

Denote the inverse of \(A_2\) by \(A_2^{-1} := [b_1 \ b_2]\), where \(b_1\) and \(b_2\) are \(2 \times 1\) vectors. By \(A_2 A_2^{-1} = I\), we readily have

\[
\begin{align*}
e_1^T b_1 &= \frac{\theta - 1}{\eta - 1}, \\
e_2^T b_1 &= 0, \\
(\eta e_1^T + e_2^T) b_1 &= 0, \\
(\eta e_1^T + e_2^T) b_2 &= \lambda.
\end{align*}
\]

Therefore,

\[
\begin{align*}
A_1 A_2^{-1} A_3 &= \begin{bmatrix}
\frac{\theta e_1^T}{\eta - 1} + e_2^T \\
\frac{\theta e_1^T}{\eta - 1} + e_2^T
\end{bmatrix} \begin{bmatrix}
b_1 \\
b_2
\end{bmatrix} A_3 \\
&= \begin{bmatrix}
\frac{\theta e_1^T}{\eta - 1} + e_2^T \\
\frac{\theta e_1^T}{\eta - 1} + e_2^T
\end{bmatrix} \begin{bmatrix}
\frac{\theta e_1^T}{\eta - 1} + e_2^T \\
\frac{\theta e_1^T}{\eta - 1} + e_2^T
\end{bmatrix}
\]

(36)

With \(p = \frac{\mu - 1}{\mu - \lambda}\), we have

\[
(p - 1) A_1 + A_3 - p A_1 A_2^{-1} A_3 = 0.
\]

(37)

Hence, \(\text{rank}((A_1^{-1} A_3 - I) - p (A_2^{-1} A_3 - I)) = \text{rank}((p - 1) A_1 + A_3 - p A_1 A_2^{-1} A_3) = 1.\)

**Lemma III.8.** For two vectors \(v_1 = c_1 e_1 + d_1 e_2\) and \(s_2 = c_2 e_1 + d_2 e_2\), where \(c_1, d_1, c_2, d_2\) are elements over the finite field \(\mathbb{F}_q\) and \(c_1 d_2 - c_2 d_1 \neq 0\), then we have \(v_1^T \omega = 0\) and \(v_2^T \omega \neq 0\) with \(\omega = c_1 e_2 - d_1 e_1\).

**Proof.** This can be shown by direct calculations:

\[
\begin{align*}
v_1^T \omega &= (c_1 e_1^T + d_1 e_2^T) (c_1 e_2 - d_1 e_1) \\
&= -c_1 d_1 + d_1 c_1 = 0,
\end{align*}
\]

(38)

\[
\begin{align*}
v_2^T \omega &= (c_2 e_1^T + d_2 e_2^T) (c_1 e_2 - d_1 e_1) \\
&= c_1 d_2 - c_2 d_1 \neq 0.
\end{align*}
\]

**Lemma III.9.** There exist \(\varphi_{51}'\) and \(\varphi_{52}'\) satisfying (28).

**Proof.** Let \(c_1 = \frac{-\lambda (\mu - 1)}{\mu (\mu - \lambda) (\theta - 1)},\)

\[
d_1 = \frac{1}{\mu} + \frac{\mu - 1}{\mu - \lambda} \frac{\theta - 1}{\theta - 1},\quad c_2 = \eta \quad \text{and} \quad d_2 = 1.
\]

By (6a)–(6c) and (6j), we can derive \(c_1 d_2 - c_2 d_1 \neq 0\). Let \(v_1 = c_1 e_1 + d_1 e_2, v_2 = c_2 e_1 + d_2 e_2\). In the proof of Lemma III.7, we show that when \(p = \frac{\mu - 1}{\mu - \lambda}\),

\[
(p - 1) A_1 + A_3 - p A_1 A_2^{-1} A_3 = \begin{bmatrix}
v_1^T \\
0^T
\end{bmatrix},
\]

(40)

It follows from Lemma III.8 that if \(c_1 d_2 - c_2 d_1 \neq 0\), then \(\omega = c_1 e_2 - d_1 e_1\) satisfies \(v_1^T \omega = 0\) and \(v_2^T \omega \neq 0\). Let \(\varphi_{52} = \omega\), then \(v_1^T \varphi_{52} = v_1^T \omega = 0\) and \(v_2^T \varphi_{52}' = v_2^T \omega \neq 0\). Hence, \(\varphi_{52}\) is in fact a non-zero solution for (31) since

\[
((A_1^{-1} A_3 - I) - p (A_2^{-1} A_3 - I)) \varphi_{52} = 0.
\]

(41)
Let $\varphi'_{51} = A_3^{-1}(A_3 - A_1)\varphi_{52}$. Then the first two equalities in (28) are met. Let $c_2 = 0$, $d_2 = 1$, and $v_2 = c'_2 e_1 + d'_2 e_2 = e_2$. By (6a)-(6c), we derive

$$c_1 d'_2 - c'_2 d_1 = c_1 = \frac{-\lambda \theta (\mu - 1)}{\mu (\mu - \lambda)(\theta - 1)} \neq 0 \quad (42)$$

Again, Lemma III.8 implies that $\omega = c_1 e_2 - d_1 e_1$ satisfies $v_2^T \omega = e_2^T \varphi_{52} \neq 0$, as $c_1 d'_2 - c'_2 d_1 \neq 0$. Since we also have $A_1^T = \mu A_3^T + e_2 0$ in (5) and $v_2^T \varphi_{52} = v_2^T \omega \neq 0$, it readily follows.

$$\begin{align*}
\text{rank}(\begin{bmatrix} \varphi'_{51}^T \\ \varphi'_{52} \end{bmatrix}) &= \text{rank}(\begin{bmatrix} \varphi'_{51}^T (A_3 - A_1)^T (A_1^T)^{-1} \\ \varphi'_{52}^T \\ \varphi'_{52}^T A_1^T \\ 0 \\ v_2^T x \end{bmatrix}) \\
&= \text{rank}(\begin{bmatrix} \varphi'_{51}^T (A_3 - A_1)^T (A_1^T)^{-1} \\ \varphi'_{52}^T \\ \varphi'_{52}^T A_1^T \\ \varphi'_{52}^T v_2 \\ 0 \end{bmatrix}) \\
&= \text{rank}(\begin{bmatrix} \varphi'_{51}^T (A_3 - A_1)^T (A_1^T)^{-1} \\ \varphi'_{52}^T \\ \varphi'_{52}^T A_1^T \\ \varphi'_{52}^T v_2 \\ 0 \end{bmatrix}) \\
&= 2;
\end{align*}$$

i.e., the last equality in (28) is satisfied

Theorem III.10. The second parity node can be optimally repaired.

Proof. By Lemma III.9, we have found proper $\varphi'_{51}$ and $\varphi'_{52}$ satisfying (28). Hence, through downloading one symbol from each surviving node and performing interference alignment, the second parity node can be optimally repaired.

Fig. 6 illustrates the optimal repair of the second parity node, with the proposed MDS codes constructed over the finite field $\mathbb{F}_4$ in Tab. II. The operations are similar to those with repair of the first parity node.

D. Discussions

Theorems III.4, III.6, and III.10 together establish that the proposed MDS codes have optimal repair capability; i.e., they can ensure exact repair of every single node failure using minimum bandwidth. With the proposed MDS codes, the node repairs require low-complexity computations. As shown in Fig. 2 to Fig. 6, the newcomer downloads one linear combination of the $\alpha = 2$ symbols from each of the $n - 1$ surviving nodes; the relevant computation complexity is $O((n - 1)\alpha)$. The interference cancellation requires a computational complexity $O((k - 1)\alpha/2)$. After canceling out all the interference, the desired data can be repaired by multiplying an $\alpha \times \alpha$ matrix, with a computation complexity $O(\alpha^2)$ for repairing systematic nodes, and $O(M \alpha)$ for repairing parity nodes. Since the proposed codes have a minimum stripe size $\alpha = 2$ and are constructed over $\mathbb{F}_4$, the computational complexity with node repairs is clearly very low.

Recall that the proposed MDS codes can be only used to store a data file of $M = 6$ symbols. However, following the simple duplication technique in [2], we can employ the proposed codes to store a data file with more symbols. For example, we first construct two DSS (denoted by DSS0 and DSS1) using the proposed MDS codes, and then combine them into a new DSS (denoted by DSS2). The first node of DSS0 contains two symbols denoted by $m_{11}$ and $m_{12}$, and the first node of DSS1 contains two symbols denoted by $m'_{11}$ and $m'_{12}$, then the first node of DSS2 contains four symbols: $m'_{11}$, $m_{12}$, $m_{11}$, and $m'_{12}$. Constructions of other nodes in DSS2 are similar. By this way, we can double the storage capacity of DSS and all the properties remains. Assume two nodes fail in DSS2, we can still retrieve the original file using the surviving nodes. Because the symbols in these nodes are able to retrieve the original file of DSS0 and DSS1 respectively, they can also retrieve the original file of DSS2, which is the combination of files in DSS0 and DSS1. Similarly, optimal node repairs can be also performed after duplication.

IV. CONCLUSIONS

We developed a novel construction of $(5, 3)$ MDS codes. Constructed over a small finite field $\mathbb{F}_4$, our codes have the minimum stripe size $\alpha = 2$, and ensure optimal repair of every single node failure using minimum bandwidth. It is certainly interesting to extend our approach to construct general $(k + 2, k)$ MDS codes with small stripe size and optimal repair capability. This will be pursued in future work.

REFERENCES

[1] A. Dimakis, P. Godfrey, Y. Wu, M. Wainwright, and K. Ramchandran, “Network coding for distributed storage systems,” IEEE Trans. Inf. Theory, vol. 56, no. 9, pp. 4539–4551, Sep. 2010.
[2] K. V. Rashmi, N. B. Shah, and P. V. Kumar, “Optimal exact-repairing codes for distributed storage at the MSR and MBR points via a product-matrix construction,” IEEE Trans. Inf. Theory, vol. 57, no. 8, pp. 5227–5239, Aug. 2011.
[3] Y. Wu, “A construction of systematic MDS codes with minimum repair bandwidth,” IEEE Trans. Inf. Theory, vol. 57, no. 6, pp. 3738–3741, Jun. 2011.
[4] 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. Inf. Theory, vol. 58, no. 4, pp. 2134–2158, Apr. 2012.
[5] C. Suh and K. Ramchandran, “Exact-repair MDS code construction using interference alignment,” IEEE Trans. Inf. Theory, vol. 57, no. 3, pp. 1425–1442, Mar. 2011.
[6] S. Liu, C. Yuan, and H. Kan, “Linear exact-repair construction of hybrid MMR codes in distributed storage systems,” IEEE Commun. Letters, vol. 18, no. 7, pp. 1095–1098, Jul. 2014.
[7] Y. Wu and A. Dimakis, “Reducing repair traffic for erasure coding-based storage via interference alignment,” in Proc. ISIT, pp. 2276–2280, Seoul, South Korea, Jul. 2009.
[8] D. S. Papailiopoulos, A. G. Dimakis, and V. Cadambe, “Repair optimal erasure codes through Hadamard designs,” IEEE Trans. Inf. Theory, vol. 59, no. 5, pp. 3021–3037, May 2013.
[9] Z. Wang, I. Tamo, and J. Bruck, “Long MDS codes for optimal repair bandwidth,” in Proc. ISIT, July 2012.
[10] I. Tamo, Z. Wang, and J. Bruck, “Zigzag codes: MDS array codes with optimal rebuilding,” IEEE Trans. Inf. Theory, vol. 59, no. 3, pp. 1597–1616, Mar. 2013.
[11] Y. Wang, X. Yin, and X. Wang, “MDC codes: A new class of raid-6 codes with optimal rebuilding and encoding,” IEEE J. Sel. Areas Commun., vol. 32, no. 5, pp. 1008–1018, 2014.
[12] V. Cadambe and S. Jafar, “Interference alignment and spatial degrees of freedom for the k user interference channel,” IEEE Trans. Inf. Theory, vol. 54, no. 8, pp. 3425–3441, Aug. 2008.
[13] M. Maddah-Ali, A. Motahari, and A. Khandani, “Communication over MIMO X channels: Interference alignment, decomposition, and performance analysis,” IEEE Trans. Inf. Theory, vol. 54, no. 8, pp. 3457–3470, Aug. 2008.