

# Access vs. Bandwidth in Codes for Storage

Itzhak Tamo<sup>†</sup>, Zhiying Wang\* and Jehoshua Bruck\*

\*Electrical Engineering Department, California Institute of Technology, Pasadena, CA 91125, USA

<sup>†</sup>Dept. of ECE and Inst. for Systems Research University of Maryland, USA

tamo@umd.edu, zhiying@caltech.edu, bruck@caltech.edu

## Abstract

Maximum distance separable (MDS) codes are widely used in storage systems to protect against disk (node) failures. A node is said to have capacity  $l$  over some field  $\mathbb{F}$ , if it can store that amount of symbols of the field. An  $(n, k, l)$  MDS code uses  $n$  nodes of capacity  $l$  to store  $k$  information nodes. The MDS property guarantees the resiliency to any  $n - k$  node failures. An *optimal bandwidth* (resp. *optimal access*) MDS code communicates (resp. accesses) the minimum amount of data during the repair process of a single failed node. It was shown that this amount equals a fraction of  $1/(n - k)$  of data stored in each node. In previous optimal bandwidth constructions,  $l$  scaled polynomially with  $k$  in codes with asymptotic rate  $< 1$ . Moreover, in constructions with a constant number of parities, i.e. rate approaches 1,  $l$  is scaled exponentially w.r.t.  $k$ . In this paper, we focus on the later case of constant number of parities  $n - k = r$ , and ask the following question: Given the capacity of a node  $l$  what is the largest number of information disks  $k$  in an optimal bandwidth (resp. access)  $(k + r, k, l)$  MDS code. We give an upper bound for the general case, and two tight bounds in the special cases of two important families of codes. Moreover, the bounds show that in some cases optimal-bandwidth code has larger  $k$  than optimal-access code, and therefore these two measures are not equivalent.

## I. INTRODUCTION

Erasure-correcting codes are the basis for widely used storage systems, where disks (nodes) correspond to symbols in the code. An important family of codes is the Maximum distance separable (MDS) codes, which provide an optimal resiliency to erasures for a given amount of redundancy. Namely, an MDS code with  $r$  redundancy (parity) symbols can repair the information from any  $r$  symbol erasures. Because of this storage efficiency, MDS codes are highly favorable, and a lot of research has been done to construct them. Examples of MDS codes are the well known Reed Solomon codes, EVENODD [1], [2], B-code [24], X-code [25], RDP [7], and STAR-code [9]. It is evident that in the case of  $r$  erasures, one needs to communicate all the surviving information during the repair process. However, although the MDS codes used in practice are resilient to more than a single erasure, i.e. number of parity nodes  $r > 1$ , the practical and more interesting question is; what is the minimum repair bandwidth in a single node erasure. The repair bandwidth is defined as the amount of information communicated during the repair process. This question has received much interest recently due to both its practical and theoretical importance. From a practical viewpoint, decreasing the repair bandwidth shortens both the repair process and the inaccessibility time of the erased information. Moreover, from a theoretical perspective, this question has deep connections to the widely used interference alignment technique and network coding.

### A. The Problem

The problem of efficient repair was defined by Dimakis et al. in [8]. It considers a file of size  $M$  symbols, divided into  $k$  equally sized chunks stored using an  $(n, k, l)$  MDS code over the finite field  $\mathbb{F}$ , where  $n$  is the number of nodes, each of capacity  $l = \frac{M}{k \log |\mathbb{F}|}$ . Namely, each node can store up to  $l$  symbols and each symbol corresponds to  $\log |\mathbb{F}|$  bits. The first  $k$  nodes, which are referred to as the systematic nodes, store the raw information. The later  $r = n - k$  nodes are the parity nodes which store a function of the raw information. Since the code is MDS, it can tolerate *any* loss of up to  $r$  nodes. However, the more common scenario is the failure (erasure) of only one node. [8] proved that

$$l \cdot \frac{n-1}{n-k} \tag{1}$$

is a lower bound on the repair bandwidth for an  $(n, k, l)$  MDS code. For example, in a code with  $r = 2$  parities, each of the  $n - 1$  surviving nodes needs to communicate during the repair process, on the average at least  $l/2$  symbols, which is equal to one half of the node's capacity. Note that repair is possible since the code is resilient to more than one erasure, and a repair strategy of communicating the entire remaining information suffices. An MDS code is termed *optimal bandwidth* if it achieves the lower bound in (1) during the repair process of any of its systematic nodes<sup>1</sup>. Figure 1 shows an optimal bandwidth  $(6, 4, 2)$  MDS code. For repairing an erased node, one symbol of information is transmitted to the repair center from each surviving node. In some applications such as data centers, reading (accessing) the information is more costly than transmitting

The material in this paper was presented in part at the IEEE International Symposium on Information Theory (ISIT 2012), Cambridge, MA, USA, July 2012.

<sup>1</sup>The relaxed requirement of optimal repair only for the systematic nodes is reasonable, because the number of parity nodes in most storage systems is negligible compared to systematic nodes. Moreover, in an erasure of a systematic node, the raw information is not accessible as opposed to a parity node erasure.

N1	N2	N3	N4	Parity 1	Parity 2
a	b	c	d	a+b+c+d	a+5w+b+2c+5d
w	x	y	z	w+x+y+z	3w+2b+3x+4y+5z

**Figure 1.** An  $(6,4,2)$  MDS code with optimal bandwidth over the field  $\mathbb{F}_7$ . Nodes  $N_1, N_2, N_3, N_4$  are systematic and the last 2 nodes are parity nodes. For repairing node  $N_1$ , (resp.  $N_2$ ) transmit the first (second) row from each surviving node. For repairing node  $N_3$  transmit from each surviving node the sum of its two elements. For repairing node  $N_4$  transmit the sum of the first row and twice the second row from Parity 2, and the sum of the first row and four times the second row from the rest. Notice that this code can be converted to be over the field of size 4, i.e. an  $(6,4,2)$  MDS code with optimal bandwidth over the field  $\mathbb{F}_2$ .

	Optimal Bandwidth	Optimal Access
Optimal update	$k = \log_r l, \checkmark *^2$ [17]	$k = \log_r l, \checkmark$ [17]
Non-Optimal update	$(r+1) \log_r l \leq k \leq l \binom{l}{l/r}, *$ [20]	$k = r \log_r l, \checkmark *$ [4]

**Figure 2.** Summary of known results on the maximum number of information nodes  $k$  in an  $(k+r, k, l)$  MDS code. The derived upper bounds apply for codes with constant repairing subspaces. The upper bounds in the general case (not necessarily constant repairing subspaces) are at most greater by one than the bounds presented in the table.  $\checkmark$  indicates a tight bound,  $*$  indicates a new upper bound. The references refer to previously known lower bounds

it. Therefore during a repair process, the need to transmit data that is a function of a large portion of the information stored within a node, can cause a bottleneck. For example, node  $N_1$  needs to access its entire stored information, for it to calculate  $a+w$ , during the repair process of node  $N_3$ . Therefore, in a large scale storage systems, one might need to minimize not only the amount of information transmitted but also the number of accessed information elements. An *optimal access* MDS code is an optimal bandwidth code that transmits only the elements it accesses. By definition, any optimal access code is also an optimal bandwidth code. The shortened code restricted to nodes  $\{N_1, N_2, \text{Parity 1}, \text{Parity 2}\}$  in Figure 1 is an example of an optimal access  $(4,2,2)$  MDS code. In [15] a similar scheme termed *repair by transfer* was considered. In this scheme an exact repair of a lost node is performed by mere transmission of information, without any calculation in any of the surviving nodes *or* at the repair center.

In a value's update of a stored element, one needs to update each parity node at least once. To avoid an overload on the system during a frequent operation such as updating, one needs to design an *optimal update* code, that updates exactly once in each parity node, when an element changes its value. For example in Figure 1 the shortened code restricted to nodes  $\{N_3, N_4, \text{Parity 1}, \text{Parity 2}\}$  is an optimal update and optimal bandwidth  $(4,2,2)$  MDS code, because updating any of the elements  $c, d, y, z$  will require updating exactly one element in each of the parity nodes.

Various codes [5], [8], [12]–[14], [16], [21]–[23] were constructed with the goal of achieving optimal bandwidth, however these constructions all have low rate, i.e.,  $k/n \leq 1/2$ . In [14], [16], [22] the key idea was using vector coding. Namely, each symbol in a codeword is a vector and not scalar as in “standard” codes. Specifically [14], [16] constructed optimal bandwidth  $(2k, k, k)$  MDS codes. Using interference alignment, it was shown in [6] that the bound in (1) is asymptotically achievable also for high rate codes ( $k/n \geq 1/2$ ). The question of existence of optimal bandwidth codes with high rate was resolved in several constructions [3], [4], [10], [11], [17]–[19]. The constructions have an arbitrary number of parity nodes  $r$ , however when  $r$  is constant, i.e. rate approaching 1 in all of the constructions  $k = O(\log_r l)$ , i.e., the capacity  $l$  scales exponentially with the number of systematic nodes  $k$ .

## B. Our Contribution

Our main goal in this paper is to understand the relation between  $l$  the capacity of each node, and the number of systematic nodes  $k$ . More precisely, given the capacity of the node  $l$ , what is the largest number of systematic nodes  $k$ , such that there exists an *optimal bandwidth* or *optimal access*  $(k+r, k, l)$  MDS code, for some constant  $r$ . We will derive three upper bounds on the number of nodes  $k$  as a function of *only*  $l$ , for different families of codes. We emphasize that we consider *only* linear codes, and the bounds apply for this case only. To derive the bounds, we use three different combinatorial techniques. The first bound considers the general problem, where no requirements on the MDS code are imposed except the optimal bandwidth property. The bound is derived by defining an appropriate set of multivariate polynomials. We proceed by deriving a *tight* bound for optimal bandwidth MDS codes with diagonal encoding matrices. These codes are a part of an important family of codes with an *optimal update* property. The last result provides a *tight* bound on *optimal access* MDS codes. Table 2 summarizes the known results together with our new results.

For constant  $r$ , all the previous optimal-bandwidth constructions [3], [4], [10], [11], [17]–[19] are indeed either optimal-access codes or equivalent to optimal-access codes. Therefore, it is not obvious whether there can be any difference between these two kinds of optimality. From the second row of Table 2, we discovered that for fixed  $l$  and  $r$ , the maximum possible number of systematic nodes are not the same for an optimal-bandwidth and an optimal-access code. That is to say, these two criteria of optimality are not equivalent when a code is non-optimal update.

<sup>2</sup>The result we present considers a special case of optimal update code, where the encoding matrices are diagonal.

An example of the size of a practical code can be as follows. In today's current technology the size of an ordinary disk in large storage systems is approximately 1TB =  $2^{40}$  bits. Hence, each node stores at most  $2^{40}$  symbols. Applying for example the upper bound in the table for optimal access codes we get that there are at most  $2 \cdot \log 2^{40} = 80$  nodes in the system.

The remainder of the paper is organized as follows. Section II presents the settings of the problem and some notation. Section III provides an upper bound for the most general case, i.e., an MDS code with optimal bandwidth property. We proceed in Section IV where a bound is derived for codes with diagonal encoding matrices. In Section V a bound for codes with optimal access property is derived. We conclude with a summary in Section VI .

## II. SETTINGS AND NOTATION

Consider a file of size  $\mathcal{M} = kl$ , divided into  $k$  nodes of capacity  $l$  over the field  $\mathbb{F}$ , namely each node can store up to  $l$  elements of that field. Each systematic node  $1 \leq i \leq k$  is represented by an  $l \times 1$  vector  $a_i \in \mathbb{F}^l$ . Interchangeably, we will refer to a matrix  $S$  and the subspace spanned by its rows as the same mathematical object, therefore

$$\text{rank}(S) = \dim(S).$$

Moreover, whenever we write an equality between two matrices we mean to an equality between the subspaces spanned by their rows. For any integer  $r$  an  $(k+r, k, l)$  MDS code is constructed by adding parity nodes  $k+1, \dots, k+r$ , which will give the resiliency to node erasures. Parity node  $k+i$  for  $i \in \{1, \dots, r\}$  stores the information vector  $a_{k+i}$  of length  $l$  over  $\mathbb{F}$ , and is defined as

$$a_{k+i} = \sum_{j=1}^k C_{i,j} a_j.$$

Here the  $C_{i,j}$ 's are invertible matrices of order  $l$ , which are called the encoding matrices. Note that the code has a systematic structure, i.e., the first  $k$  nodes store the information itself, and not a function of it. Therefore, the code is uniquely defined by the matrix

$$C = (C_{i,j})_{i \in [r], j \in [k]} = \begin{bmatrix} C_{1,1} & \dots & C_{1,k} \\ \vdots & \ddots & \vdots \\ C_{r,1} & \dots & C_{r,k} \end{bmatrix}. \quad (2)$$

The code is called an MDS if it can repair any  $r$  node erasures, which is equivalent to the statement that any  $1 \times 1, 2 \times 2, \dots, r \times r$  block sub matrix in (2) is invertible. Consider a scenario of a single erasure of a systematic node  $m$ ,  $1 \leq m \leq k$ . In order to optimally repair the lost data, a linear combination of the information stored in the parity nodes is transmitted to the erased node. Namely, parity nodes  $k+1, \dots, k+r$ , project their data on the repairing subspaces  $S_{1,m}, S_{2,m}, \dots, S_{r,m}$  of dimension  $l/r$  each, respectively. During the repair process of systematic node  $m \in [k]$ , parity node  $k+i$  transmits the information

$$S_{i,m} a_{k+i} = S_{i,m} \sum_{j=1}^k C_{i,j} a_j.$$

The *only* information about the lost systematic node  $m$  received by parity node  $k+i$  is  $S_{i,m} C_{i,m} a_m$ . Note that the other surviving systematic nodes *do not* contain any information about the lost node. Therefore a necessary condition for repairing the lost information of systematic node  $m$  is

$$\text{rank} \begin{bmatrix} S_{1,m} C_{1,m} \\ \vdots \\ S_{r,m} C_{r,m} \end{bmatrix} = l, \quad (3)$$

i.e., the matrix is invertible. This condition is equivalent to that the subspaces  $S_{1,m} A_{1,m}, \dots, S_{r,m} A_{r,m}$  form a direct sum of  $\mathbb{F}^l$ , namely

$$\bigoplus_{i \in [r]} S_{i,m} C_{i,m} = \mathbb{F}^l. \quad (4)$$

However the transmitted information from the parities contains interference (information) from the other surviving nodes. The interference of node  $m' \neq m$  received from parity node  $k+i$  is  $S_{i,m} C_{i,m'} a_{m'}$ . Systematic node  $m'$  transmits to the repair center enough information in order to cancel out the this interference. In total, the information that needs to be transmitted from node  $m'$  is

$$\begin{bmatrix} S_{1,m} C_{1,m'} \\ \vdots \\ S_{r,m} C_{r,m'} \end{bmatrix} a_{m'}. \quad (5)$$

Hence the amount of information transmitted is equivalent to the rank of the matrix in (5). The rank of the matrix  $S_{1,m}C_{1,m'}$  is  $l/r$ , therefore the rank of the whole matrix is at least  $l/r$ . Thus the code is optimal bandwidth only if we transmit the smallest amount of information, i.e. for any  $m' \neq m$

$$\text{rank} \begin{bmatrix} S_{1,m}C_{1,m'} \\ \vdots \\ S_{r,m}C_{r,m'} \end{bmatrix} = \frac{l}{r}. \quad (6)$$

Which is equivalent to the equality between the subspaces

$$S_{1,m}C_{1,m'} = S_{2,m}C_{2,m'} = \dots = S_{r,m}C_{r,m'}. \quad (7)$$

We conclude that an optimal bandwidth algorithm for the systematic nodes is defined by the set of repairing subspaces  $(S_{1,m}, \dots, S_{r,m})$  that satisfy (3) and (6) for  $1 \leq m \leq k$ .<sup>3</sup> However, it will be more convenient to assume that the repairing subspaces are constant, namely to repair systematic node  $m$  we use the same repairing subspace  $S_m$  for each of the  $r$  parities. In other words, the information transmitted from parity node  $k+i$  is  $S_m a_{k+i}$ . From Combining equations (3), (6) we get the following corollary.

**Corollary 1** *The code defined in (2) is optimal bandwidth with constant repairing subspaces if there exist subspaces  $S_1, \dots, S_k$  each of dimension  $l/r$ , such that for any  $m \in [k]$*

$$\text{rank} \begin{bmatrix} S_m C_{1,m'} \\ \vdots \\ S_m C_{r,m'} \end{bmatrix} = \begin{cases} l & m = m' \\ l/r & \text{else,} \end{cases}. \quad (8)$$

The following remarks apply for codes with constant repairing subspaces.

**Remarks:**

- 1) Without loss of generality we will always assume that the last row in the encoding matrix  $\mathcal{C}$  in (2) is composed of only identity matrices, i.e.,  $C_{r,m} = I$  for any  $m \in [k]$ . Because if  $\mathcal{C} = (C_{i,j}), i \in [r], j \in [k]$  defines an optimal bandwidth code, let  $C'_{i,j} = C_{i,j}C_{r,j}^{-1}$ . Then  $\mathcal{C}' = (C'_{i,j}), i \in [r], j \in [k]$  with the same sets of repairing subspaces, defines an optimal bandwidth code, and  $C'_{r,m}$  is the identity matrix for any  $m \in [k]$ .
- 2) Since the dimension of each subspace  $S_m$  is  $l/r$ , and any encoding matrix  $C \in \{C_{i,j}\}$  is invertible, then  $\dim(S_m C) = l/r$ . Hence the rank of the matrix in (8), which is composed of  $r$  block matrices, has two extreme cases for its possible value. For  $m = m'$  the rank is maximal, i.e. the matrix is invertible. For  $m \neq m'$  the rank has the minimum possible value of  $l/r$ . Note also that in this case, for any  $i \in [r]$

$$S_m C_{i,m'} = S_m. \quad (9)$$

Namely  $S_m$  is an invariant subspace for any matrix  $C_{i,m'}$  when  $m' \neq m$ . This follows since  $C_{r,m'}$  is assumed to be the identity matrix according to the previous remark.

- 3) For  $m' = m$  (8) is equivalent to

$$\bigoplus_{i \in [r]} S_m C_{i,m} = \mathbb{F}^l. \quad (10)$$

The next theorem shows that from any optimal bandwidth MDS code we can construct another optimal bandwidth MDS code with constant repairing subspaces, and almost the same parameters.

**Theorem 2** *If there exists an optimal bandwidth  $(k+r, k, l)$  MDS code, then there exists an optimal bandwidth  $(k+r-1, k-1, l)$  MDS code with constant repairing subspaces.*

The proof is shown in Appendix A.

From the last theorem we get the following corollary.

**Corollary 3** *Let  $k$  be the largest number of systematic nodes in an optimal bandwidth  $(k+r, k, l)$  MDS code. Let  $s$  be the largest number of systematic nodes in an optimal bandwidth  $(s+r, s, l)$  MDS code with constant repairing subspaces, then  $s \leq k \leq s+1$ .*

*Proof:* It is clear that  $s \leq k$ . From Theorem 2 we conclude that  $k-1 \leq s$ . ■

Theorem 2 shows that the difference between the maximum number of nodes  $k$  in an optimal bandwidth MDS codes with or without constant repairing subspaces is negligible (at most 1). Therefore in the sequel we will always assume that the codes have constant repairing subspaces, and the bounds will apply for this case.

For any two integers  $i < j$  denote by  $[i] = \{1, \dots, i\}$  and  $[i, j] = \{i, i+1, \dots, j\}$ . For simplicity, we will assume that the capacity of each node  $l$ , is a power of  $r$ . In the next section we present our first bound which applies for the most general case.

<sup>3</sup>We point out that similar conditions were derived also in [14].

### III. UPPER BOUND ON THE NUMBER OF NODES IN AN OPTIMAL BANDWIDTH MDS CODE

We start with the most general problem, which seems to be the most difficult. No constraints on the encoding matrices and the repairing subspaces are imposed. We derive an upper bound on the number of information nodes  $k$  in an optimal bandwidth  $(k+r, k, l)$  MDS code for arbitrary number of parities  $r$ . The bound is a function of *only* the capacity  $l$  of the node, regardless of the field size being used.

Before we prove the upper bound, for a set of indices  $I, J$  define  $B_{I,J}$  to be the sub matrix of  $B$  restricted to rows  $I$  and columns  $J$ .

**Theorem 4** Let  $\mathcal{C} = (C_{i,j})$  be an  $(k+r, k, l)$  optimal bandwidth MDS code with constant repairing subspaces  $S_1, \dots, S_k$  then

$$k \leq l \binom{l}{l/r}.$$

*Proof:* By the optimal bandwidth property, for any  $m \in [k]$  the matrix

$$\begin{pmatrix} S_m C_{1,m} \\ \vdots \\ S_m C_{r,m} \end{pmatrix}, \quad (11)$$

is of full rank. Here  $S_m$  is a matrix of dimension  $\frac{l}{r} \times l$ . Hence there exists a set of indices  $I \subset [l]$  of size  $\frac{l}{r} + 1$  such that the  $(\frac{l}{r} + 1) \times (\frac{l}{r} + 1)$  sub matrix restricted to rows  $[l(r-1)/r, l]$  and columns  $I$ , is invertible. Namely,

$$\det \begin{pmatrix} S_m C_{1,m} \\ \vdots \\ S_m C_{r,m} \end{pmatrix}_{[l(r-1)/r, l], I} \neq 0.$$

Moreover, since for any  $m' \neq m$ ,

$$\text{rank} \begin{pmatrix} S_m C_{1,m'} \\ \vdots \\ S_m C_{r,m'} \end{pmatrix} = \frac{l}{r},$$

the sub matrix restricted to the same set of rows and columns is not of full rank, (note that for distinct  $m$ 's the set of indices  $I$  might be different). Hence, for each  $m \in [k]$  the polynomial  $f_m : \mathbb{F}^{\frac{l}{r} \times l} \rightarrow \mathbb{F}$ , defined by,

$$f_m(S) = \det \begin{pmatrix} S C_{1,m} \\ \vdots \\ S C_{r,m} \end{pmatrix}_{[l(r-1)/r, l], I}, \quad (12)$$

satisfies,

$$f_m(S_{m'}) = \begin{cases} \neq 0 & m = m' \\ 0 & \text{otherwise.} \end{cases} \quad (13)$$

We claim that the  $f_m$ 's are linearly independent multivariate polynomials. Assume that for some  $\alpha_m$ 's  $\in \mathbb{F}$

$$\sum_m \alpha_m f_m = \vec{0},$$

where  $\vec{0}$  is the zero polynomial. Assume by contradiction that  $\alpha_j \neq 0$  for some  $j$ , but

$$\begin{aligned} 0 &= \vec{0}(S_j) \\ &= \sum_m \alpha_m f_m(S_j) \\ &= \alpha_j f_j(S_j) \neq 0, \end{aligned}$$

and we get a contradiction. Therefore the polynomials are linearly independent. Define two sets of polynomials

$$T_1 = \left\{ \det \begin{pmatrix} x_{1,1} & \cdots & x_{1,l} \\ \vdots & \ddots & \vdots \\ x_{\frac{l}{r},1} & \cdots & x_{\frac{l}{r},l} \end{pmatrix} : J \in \binom{[l]}{\frac{l}{r}} \right\},$$

and  $T_2 = \{x_{l/r,i} : 1 \leq i \leq l\}$ , where  $\binom{[l]}{l/r}$  is the set of  $l/r$ -subsets of  $[l]$ . Note that each element in the  $l(r-1)/r$ -th row of (11) is a linear combination of the indeterminates  $x_{l/r,1}, \dots, x_{l/r,l}$  in the last row. In addition, recall that  $C_{r,m}$  is the identity matrix and  $S_m C_{r,m} = S_m$ . Hence, by expanding the determinant in (12) by the  $l(r-1)/r$ -th row, we conclude that it is a linear combination of the polynomials from

$$T_1 \cdot T_2 = \{h \cdot g : h \in T_1, g \in T_2\}.$$

Namely,  $\{f_m\} \subseteq \text{span}(T_1 \cdot T_2)$ . However, since the  $f_m$ 's are linearly independent, the number of polynomials is at most the dimension, i.e.,

$$\begin{aligned} k &= |\{f_m\}| \\ &\leq \dim(\text{span}(T_1 \cdot T_2)) \\ &\leq |T_1| \cdot |T_2| \\ &= l \binom{l}{l/r}. \end{aligned}$$

**Corollary 5** *Let  $k$  be the largest number of systematic nodes in an optimal bandwidth  $(k+r, k, l)$  MDS code, then*

$$(r+1) \log_r l \leq k \leq l \binom{l}{l/r}.$$

*Proof:* The lower bound is given by the code constructed in [20].

As one can notice, there exists a big gap between the upper and the lower bound. We conjecture that the lower bound is more accurate, and in fact  $k = \theta(\log l)$ .

We proceed by giving a tight bound for the number of systematic nodes  $k$  in the case where all the encoding matrices are diagonal.

#### IV. UPPER BOUND FOR DIAGONAL ENCODING MATRICES

One of the most common operation in the maintenance of a storage system is updating. Namely, a certain element has changed its value, and that needs to be updated in the system. Since the code is an MDS, each parity node is a function of the entire information stored in the system. Therefore, in a single update, each parity node needs to be updated at least in one of the elements it stores. An *optimal update* code is one that needs to update each parity node *exactly* once in an update of any information element. Namely, an optimal update code updates the minimum number of times in any value change. Since updating is a highly frequent operation, a storage system with the optimal update property has a huge advantage. A reasonable question to answer is what can be said on systems that possess both the optimal access/bandwidth and optimal update properties. In this section we derive a tight bound on the number of information disks for these systems. However the derived bound applies only for a special case of an *optimal update* code, where all the encoding matrices are diagonal. Note that in Theorem 2, if the code is composed of diagonal encoding matrices, then in the theorem, the constructed code with constant repairing subspaces will also be composed of diagonal matrices. Therefore Corollary 3 applies also to codes with diagonal matrices.

We begin with a simple lemma on the entropy function.

**Lemma 6** *Let  $X$  be a random variable such that for any possible outcome  $x$ ,  $P(X = x) \leq \frac{1}{r}$ , then its entropy satisfies  $H_r(X) \geq 1$ , where  $H_r(\cdot)$  is the entropy function calculated in base  $r$ .*

*Proof:* Since  $P(X) \leq \frac{1}{r}$  then  $\log_r(\frac{1}{P(X)}) \geq 1$  and

$$H_r(X) = E(\log_r(\frac{1}{P(X)})) \geq 1.$$

Next we make a few definitions. A partition  $\mathcal{X}$  of some set  $T$  is a set of subsets of  $T$  such that

$$\cup_{x \in \mathcal{X}} x = T,$$

and for any distinct sets  $x_1, x_2 \in \mathcal{X}$

$$x_1 \cap x_2 = \emptyset.$$

Moreover, for two partitions  $\mathcal{X}, \mathcal{Y}$ , their meet is defined as,

$$\mathcal{X} \wedge \mathcal{Y} = \{x \cap y : x \in \mathcal{X}, y \in \mathcal{Y}\}.$$



Note that the meet of two partitions of same set is also a partition. We denote partitions by Calligraphic letters  $\mathcal{A}, \mathcal{B}, \dots$ , and sets in a partition by lowercase letters, e.g.  $x \in \mathcal{X}$ . For a set of indices  $x \subseteq [l]$  denote by  $\text{span}(e_x) = \text{span}(e_i : i \in x)$ , where  $e_i$  is the  $i$ -th vector in the standard basis.

Since each encoding matrix  $C_{i,j}$  is diagonal, the standard basis vectors are its set of eigenvectors, and the entries along the diagonal are its eigenvalues. Therefore  $C_{i,j}$  defines a partition  $\mathcal{X}_{i,j}$  of  $[l]$ , by  $m, n \in [l]$  are in the same set of the partition, iff the corresponding standard basis vectors  $e_m$  and  $e_n$  have the same eigenvalue in  $C_{i,j}$ . Let  $m' \in [k]$  be some node that needs to be repaired, and denote by  $\mathcal{X}$  the meet of the partitions

$$\mathcal{X} = \wedge_{i \in [r], m \neq m'} \mathcal{X}_{i,m}.$$

In addition, let  $S = S_{m'}$  be the repair subspace for that node.

The following lemma shows that  $S$  can be decomposed into a direct sum of subspaces, such that each subspace is an invariant subspace of all the matrices  $C_{i,m}, i \in [r], m \neq m'$ . Note that for each  $x \in \mathcal{X}$  and  $m \neq m'$ , the subspace  $\text{span}(e_x)$  is a subspace of some eigenspace of  $C_{i,m}$ . Therefore,  $\text{span}(e_x)$  and  $S \cap \text{span}(e_x)$  are invariant subspaces of  $C_{i,m}$ .

**Lemma 7** *The repair subspace  $S$  of the node  $m'$  can be written as*

$$S = \bigoplus_{x \in \mathcal{X}} S_x, \quad (14)$$

where  $S_x = S \cap \text{span}(e_x)$ .

*Proof:* It is clear that a vector  $v \neq 0$  is an eigenvector for all the matrices  $C_{i,m}, m \neq m'$  iff  $v \in \text{span}(e_x)$ , for some set  $x$  in the partition  $\mathcal{X}$ . Assume  $S$  is represented in its reduced row echelon form, and without loss of generality we assume that the first  $l/r$  columns of  $S$  are linearly independent, hence

$$S = \left( \begin{array}{c|c} I_{l/r} & A \end{array} \right).$$

Here  $I_t$  is the identity matrix of order  $t$  and  $A$  is an  $l/r \times l(r-1)/r$  matrix, and recall that  $S$  is an  $l/r \times l$  matrix. For any  $j \in [l/r]$  let  $v_j = (e_j | a_j)$  be the  $j$ -th row of  $S$ , where  $a_j$  is the  $j$ -th row of  $A$ . By the optimal bandwidth property,  $S$  is an invariant subspace of any matrix  $C_{i,m}$  for any  $m \neq m'$  and  $i \in [r]$ , which are all diagonal matrices. Therefore, we get

$$v_j C_{i,m} = (\alpha e_j | a'_j) \in S = \text{span}(v_1, \dots, v_{l/r}),$$

for some non zero  $\alpha \in \mathbb{F}$  and a vector  $a'_j$ . Namely

$$\text{rank} \left( \begin{array}{c} S \\ v_j C_{i,m} \end{array} \right) = \text{rank} \left( \begin{array}{c|c} I_{l/r} & A \\ \alpha e_j & a'_j \end{array} \right) = l/r.$$

We claim that  $a'_j = \alpha a_j$ , namely  $(e_j | a_j)$  the  $j$ -th row of  $S$  is an eigenvector of  $C_{i,m}$ . This follows since since  $v_j, v_j C_{i,m} \in S$  and

$$\alpha v_j - v_j C_{i,m} = \alpha(e_j | a_j) - (\alpha e_j | a'_j) = (0 | \alpha a_j - a'_j) \in S.$$

However, the only vector in  $S$  with first  $l/r$  entries being zero, is the zero vector. Hence we conclude that  $a'_j = \alpha a_j$ , and each row vector  $v_j$  of  $S$  is an eigenvector of  $C_{i,m}$  for any  $m \neq m'$ . Namely,  $v_j \in \text{span}(e_x)$  for some set  $x$  in the partition  $\mathcal{X}$ , and the result follows.  $\blacksquare$

So far we have looked at  $\mathcal{X}$  the meet of the partitions  $\mathcal{X}_{i,m}, i \in [r], m \neq m'$ . Next, we are going to partition each set in  $\mathcal{X}$  using the partitions  $\mathcal{X}_{i,m'}, i \in [r]$ , and then upper bound the size of each set in that partition.

**Lemma 8** *For  $x \in \mathcal{X}$  denote by  $\mathcal{P}_x = x \wedge (\wedge_i \mathcal{X}_{i,m'})$ , the partition of  $x$  by  $\mathcal{X}_{i,m'}, 1 \leq i \leq r$ . Then the size of each set in the partition  $\mathcal{P}_x$  is at most  $|x|/r$ , namely*

$$\max_{z \in \mathcal{P}_x} |z| \leq \frac{|x|}{r}. \quad (15)$$

*Proof:* Assume the contrary that the size of some set  $z$  in  $\mathcal{P}_x$  is  $|z| > |x|/r$ . On one hand, for each  $x \in \mathcal{X}$  the subspace  $S_x$  is contained in  $\text{span}(e_x)$ , moreover,  $\text{span}(e_x)$  is an invariant subspace for  $C_{i,m'}$  for any  $i \in [r]$ , since it is a diagonal matrix. Therefore

$$S_x C_{i,m'} \subseteq \text{span}(e_x) C_{i,m'} = \text{span}(e_x). \quad (16)$$

In addition

$$\begin{aligned} \bigoplus_{x \in \mathcal{X}} \text{span}(e_x) &= \mathbb{F}^l \\ &= \bigoplus_{i \in [r]} S C_{i,m'} \end{aligned} \quad (17)$$

$$= \bigoplus_{i \in [r]} \bigoplus_{x \in \mathcal{X}} S_x C_{i,m'} \quad (18)$$

$$= \bigoplus_{x \in \mathcal{X}} \bigoplus_{i \in [r]} S_x C_{i,m'}. \quad (19)$$

Here (17) follows from (10) and (18) follows from (14). From (16) and (19) we conclude that for any  $x \in \mathcal{X}$

$$\bigoplus_{i \in [r]} S_x C_{i,m'} = \text{span}(e_x). \quad (20)$$

Calculating the dimensions in (20)

$$\begin{aligned} |x| &= \dim(\text{span}(e_x)) \\ &= \dim(\bigoplus_{i \in [r]} S_x C_{i,m'}) \\ &= \sum_{i=1}^r \dim(S_x C_{i,m'}) \\ &= r \dim(S_x), \end{aligned}$$

i.e.,

$$\dim(S_x) = \frac{|x|}{r}. \quad (21)$$

On the other hand, let  $\alpha_i$  be the eigenvalue of the matrix  $C_{i,m'}$  that corresponds to the vectors in  $\text{span}(e_z)$ . W.l.o.g assume that  $z = \{1, 2, \dots, |z|\}$ , hence by (20)

$$|x| = \text{rank} \begin{pmatrix} S_x C_{1,m'} \\ \vdots \\ S_x C_{r-1,m'} \\ S_x C_{r,m} \end{pmatrix} = \text{rank} \begin{pmatrix} S_x(C_{1,m'} - \alpha_1 I) \\ \vdots \\ S_x(C_{r-1,m'} - \alpha_{r-1} I) \\ S_x \end{pmatrix}. \quad (22)$$

Here the last equality in (22) follows since  $C_{r,m}$  is the identity matrix, and the two matrices are row equivalent. However, for any  $i \in [r]$ , the first  $|z|$  columns in the diagonal matrix

$$C_{i,m'} - \alpha_i I$$

are zeros. In addition  $S_x$  is contained in  $\text{span}(e_x)$ , i.e. the indices of the non zero entries in any vector of  $S_x$  are contained in  $x$ . Therefore we get that for any  $i$ ,

$$S_x(C_{i,m'} - \alpha_i I) \subseteq \text{span}(e_{x \setminus z}).$$

Hence

$$\begin{aligned} \text{rank} \begin{pmatrix} S_x(C_{1,m'} - \alpha_1 I) \\ \vdots \\ S_x(C_{r-1,m'} - \alpha_{r-1} I) \end{pmatrix} &\leq \dim(\text{span}(e_{x \setminus z})) \\ &= |x| - |z| \\ &< |x| - \frac{|x|}{r}, \end{aligned} \quad (23)$$

Therefore we have

$$\begin{aligned} |x| &= \text{rank} \begin{pmatrix} S_x C_{1,m'} \\ \vdots \\ S_x C_{r-1,m'} \\ S_x C_{r,m} \end{pmatrix} \\ &\leq \text{rank} \begin{pmatrix} S_x(C_{1,m'} - \alpha_1 I) \\ \vdots \\ S_x(C_{r-1,m'} - \alpha_{r-1} I) \end{pmatrix} + \text{rank}(S_x) \\ &< |x| - \frac{|x|}{r} + \frac{|x|}{r} \\ &= |x|. \end{aligned} \quad (24)$$

Here (24) follows from (23) and (21), therefore (15) holds. ■

Now we are ready to prove the upper bound on the number of systematic nodes.

**Theorem 9** Let  $\mathcal{C} = (C_{i,j})$  be an  $(k+r, k, l)$  optimal bandwidth code composed of diagonal encoding matrices, namely each  $C_{i,j}$  is a diagonal matrix, and constant repairing subspaces  $S_1, \dots, S_k$ , then  $k \leq \log_r l$ .



*Proof:* Let  $j$  be a random variable that gets any integer value  $1, 2, \dots, l$  with equal probability. Define for  $m' \in [k]$  the random variable  $Y_{m'}$  to be the set  $z$  in the partition  $\wedge_i \mathcal{X}_{i,m'}$  that contains  $j$ . By (15) we conclude that

$$P(Y_{m'} = z | Y_m = y_m, m \in [k] \setminus \{m'\}) \leq \frac{1}{r},$$

for any values of  $y_m, m \in [k] \setminus \{m'\}$ . Hence from Lemma 6 we conclude that the conditional entropy of  $Y_{m'}$  satisfies

$$H_r(Y_{m'} | Y_m, m \in [k] \setminus \{m'\}) \geq 1. \quad (25)$$

Therefore,

$$\begin{aligned} \log_r l &= H_r(j) \\ &= H_r(j, Y_1, \dots, Y_k) \\ &= H_r(Y_1, \dots, Y_k) + H_r(j | Y_1, \dots, Y_k) \\ &\geq H_r(Y_1, \dots, Y_k) \\ &= \sum_{m=1}^k H_r(Y_m | Y_1, \dots, Y_{m-1}) \\ &\geq \sum_{m=1}^k H_r(Y_m | Y_m, m \neq m') \end{aligned} \quad (26)$$

$$\geq \sum_{m=1}^k 1 = k, \quad (27)$$

where (26) follows since conditioning reduces entropy, and (27) follows from (25). ■

**Corollary 10** *Let  $k$  be the largest number of systematic nodes in an optimal bandwidth  $(k+r, k, l)$  MDS code with diagonal encoding matrices, then  $k = \log_r l$ .*

*Proof:* The lower bound is given by the codes constructed in [3], [10], [17], [18]. ■

Note that when restricting to diagonal encoding matrices, there is no difference if the code is an optimal access or optimal bandwidth in terms of maximum code length  $k$  (see Table 2). However, in the next section we show that these two properties are not equivalent in the general case.

## V. UPPER BOUND ON THE NUMBER OF NODES FOR OPTIMAL ACCESS

Storage systems with optimal bandwidth MDS property introduce high efficiency in data transmission during a repair process. However a major bottleneck can still emerge if the transmitted information is a function of a large portion of the data stored in each node. In the extreme case the information is a function of the *entire* information within the node. Namely, in order to generate the transmitted data from some surviving node, one has to access and read all the information stored in that node, which of course can be an expensive task. An *optimal access* code is an optimal bandwidth code that transmits only the elements it accesses. Namely, the amount of information read is equal to the amount of information transmitted. The property of *optimal access* is equivalent to that each repairing subspace  $S_i$  is spanned by an  $l/r$ -subset of the standard basis  $e_1, \dots, e_l$ , i.e.,  $S_i = \text{span}(e_m : m \in I)$  for some  $I$  an  $l/r$ -subset of  $[l]$ . As before, if the code in Theorem 2 is optimal access then the constructed code in that theorem will also have the optimal access property. This follows since the set of repairing subspaces for the newly constructed code is a subset of the repairing subspaces for the old code. Therefore Corollary 3 applies also to optimal access codes.

We start with a useful lemma that shows that in an optimal access code with constant repairing subspaces, the intersections between the subspaces are not large.

**Lemma 11** *Let  $\mathcal{C}$  be an  $(k+r, k, l)$  optimal access code with constant repairing subspaces  $S_1, \dots, S_k$ , then for any subset of indices  $T \subseteq [k]$*

$$\dim(\cap_{t \in T} S_t) \leq \frac{l}{r^{|T|}}.$$

*Proof:* We prove by induction on the size of  $T$ . For  $|T| = 1$  there is nothing to prove. For  $|T| = t$ , w.l.o.g assume that  $T = [t]$ , and denote by  $S = \cap_{j \in [t]} S_j$ . Assume the contrary that  $\dim(S) > \frac{l}{r^t}$ . It is clear by definition that  $S \subseteq S_j$  for any  $j \in [t-1]$ , hence by (9), for any  $i \in [r-1]$

$$S_{i,t} \subseteq \cap_{j \in [t-1]} S_j.$$

We conclude that  $SC_{1,t}, \dots, SC_{r,t}$  are  $r$  subspaces of dimension greater than  $l/r^t$ , which are contained in the subspace  $\cap_{j \in [t-1]} S_j$ , which by the induction hypothesis is of dimension at most  $\frac{l}{r^{t-1}}$ . Therefore the sum of these subspaces is not a direct sum, which contradicts (10). ■

**Corollary 12** *By the conditions of the previous theorem, the number of repairing subspaces  $\{S_i\}_{i=1}^k$  that contain an arbitrary vector  $v \neq 0$  is at most  $\log_r l$ .*

*Proof:* Let  $J = \{j : v \in S_j\}$ , then

$$1 \leq \dim(\cap_{j \in J} S_j) \leq \frac{l}{r^{|J|}},$$

and the result follows. ■

The previous Lemma shows that an arbitrary vector  $v \neq 0$  can not belong to “too many” repairing subspaces  $S_i$ . This observation leads to a bound on the number of nodes in an optimal access code.

**Theorem 13** *Let  $\mathcal{C}$  be an  $(k+r, k, l)$  optimal access MDS code with constant repairing subspaces  $S_1, \dots, S_k$ , then  $k \leq r \log_r l$ .*

*Proof:* Define a bipartite graph with one set of vertices to be the standard basis vectors  $e_1, \dots, e_l$ . The second set of vertices will be the repairing subspaces  $S_1, \dots, S_k$ . Define an edge between a vector  $e_i$  and a subspace  $S_j$  iff  $S_j$  contains  $e_i$ . Count in two different ways the number of edges in the graph. By the assumption the code is optimal bandwidth, hence each repairing subspace contains  $l/r$  standard basis vectors, and the degree of each repairing subspace in the graph is  $l/r$ . In total there are  $kl/r$  edges in the graph. However by Corollary 12 the degree in the graph of each standard basis vector is at most  $\log_r l$ . Hence there are at most  $l \log_r l$  edges in the graph, namely

$$k \frac{l}{r} \leq l \log_r l,$$

and the result follows. ■

**Corollary 14** *Let  $k$  be the largest number of systematic nodes in an optimal access  $(k+r, k, l)$  MDS code, then*

$$k = r \log_r l.$$

*Proof:* The lower bound is derived by the codes constructed in [4], [20]. ■

Note that [20] constructed also an optimal bandwidth code with  $k = (r+1) \log_r l$ . Therefore, in the general case where we do not require an optimal update code, there is a difference between optimal access and optimal bandwidth code. Namely, these two properties are not equivalent (see Table 2).

## VI. DISCUSSION AND SUMMARY

Assume that an MDS code over the field  $\mathbb{F}$  is to be constructed. The capacity  $l$  of each node, which is the number of symbols it can store equals to

$$l = \frac{\mathcal{M}}{\log |\mathbb{F}|},$$

where  $\mathcal{M}$  is the size in bits of the node, and  $\log |\mathbb{F}|$  is the number of bits takes to represent each symbol. In this paper we asked the following question: Given the number of parities  $r$  and the capacity  $l$ , what is the largest number of nodes  $k$  such that there exists an optimal bandwidth (resp. access)  $(k+r, k, l)$  MDS code. We used distinct combinatorial tools to derive 3 upper bounds on  $k$ . The first bound considers the general case of optimal bandwidth code. The last two bounds are tight, and they consider optimal access and optimal update codes with diagonal encoding matrices. Moreover, we showed that in the general case, the properties of optimal bandwidth and optimal access are not equivalent, although in certain codes such as codes with diagonal encoding matrices, they are. It is an open problem what is the exact bound for optimal bandwidth code with  $r$  parities and capacity  $l$ .

Since the capacity of each node is a function of the field size being used, one would like to minimize the field size in order to increase the capacity and therefore the number of nodes that can be protected. However, in order to satisfy the MDS property the field size needs to be large enough, e.g. it is well known that for optimal update codes the field  $\mathbb{F}_2$  is not sufficient. It is an interesting open problem to determine the smallest field size sufficient for the MDS property.

## VII. ACKNOWLEDGMENT

This work was partially supported by an NSF grant ECCS-0801795 and a BSF grant 2010075.

APPENDIX A  
PROOF OF THEOREM 2

**Theorem 2** *If there exists an optimal bandwidth  $(k+r, k, l)$  MDS code then there exists an optimal bandwidth  $(k+r-1, k-1, l)$  MDS code with constant repairing subspaces.*

*Proof:* Let the encoding matrices for the code in the hypothesis be

$$\begin{bmatrix} A_{1,1} & \dots & A_{1,k} \\ \vdots & \ddots & \vdots \\ A_{r,1} & \dots & A_{r,k} \end{bmatrix}, \quad (28)$$

with repairing subspaces  $(S_{1,m}, S_{2,m}, \dots, S_{r,m})$  for node  $m$ . Namely, for any distinct  $m, m' \in [k]$  the following holds

$$S_{1,m}A_{1,m'} = S_{2,m}A_{2,m'} = \dots = S_{r,m}A_{r,m'} \quad (29)$$

$$\bigoplus_{i \in [r]} S_{i,m}A_{i,m} = \mathbb{F}^l \quad (30)$$

Define the code

$$\mathcal{C} = (C_{j,m}) = \begin{bmatrix} C_{1,1} & \dots & C_{1,k-1} \\ \vdots & \ddots & \vdots \\ C_{r,1} & \dots & C_{r,k-1} \end{bmatrix},$$

where

$$C_{j,m} = A_{r,k}A_{j,k}^{-1}A_{j,m}A_{r,m}^{-1}.$$

Note that for  $C_{r,m}$  is the identity matrix for any  $m \in [k-1]$ , namely the last row in  $\mathcal{C}$  is composed of identity matrices. We claim that this is an optimal bandwidth  $(k+r-1, k-1, l)$  MDS code with constant repairing subspaces.

**Optimal Bandwidth Property:** Assume node  $m \in [k-1]$  was erased, then use the set of repairing subspaces

$$(S_m, \dots, S_m),$$

where  $S_m = S_{r,m}$ . Namely transmit from parity node  $j$  the information  $S_m a_{k+j}$ . For the optimal bandwidth property we only need to show that (8) is satisfied. Let  $m, m' \in [k-1]$  and  $j \in [r]$

$$\begin{aligned} S_m C_{j,m'} &= S_{r,m} C_{j,m'} \\ &= S_{r,m} A_{r,k} A_{j,k}^{-1} A_{j,m'} A_{r,m'}^{-1} \\ &= S_{j,m} A_{j,k} A_{j,k}^{-1} A_{j,m'} A_{r,m'}^{-1} \\ &= S_{j,m} A_{j,m'} A_{r,m'}^{-1} \end{aligned} \quad (31)$$

$$= \begin{cases} S_{j,m} A_{j,m} A_{r,m}^{-1} & m = m' \\ S_{r,m} A_{r,m'} A_{r,m'}^{-1} = S_m & \text{else,} \end{cases} \quad (32)$$

where (31) and (32) follow from (29). Therefore, for  $m' \neq m$

$$\text{rank} \begin{bmatrix} S_m C_{1,m'} \\ \vdots \\ S_m C_{r,m'} \end{bmatrix} = \text{rank} \begin{bmatrix} S_m \\ \vdots \\ S_m \end{bmatrix} = \frac{l}{r},$$

and (8) is satisfied. Moreover

$$\mathbb{F}^l = \bigoplus_{j \in [r]} S_{j,m} A_{j,m} \quad (33)$$

$$= \bigoplus_{j \in [r]} S_{j,m} A_{j,m} A_{r,m}^{-1} \quad (34)$$

$$= \bigoplus_{j \in [r]} S_m C_{j,m} \quad (35)$$

where (33) follows from (30), and (34) follows since  $A_{r,m}$  is an invertible matrix. (35) follows from (32), thus (8) is also satisfied for  $m = m'$ .

**MDS Property:** This property follows easily from the MDS code in (28). The code  $\mathcal{C}$  is MDS iff for any  $t \in [r]$  and sets of indices  $\{j_1, \dots, j_t\} \subseteq [r], \{m_1, \dots, m_t\} \subseteq [k-1]$  the block sub matrix

$$\begin{bmatrix} C_{j_1, m_1} & \dots & C_{j_1, m_t} \\ \vdots & \ddots & \vdots \\ C_{j_t, m_1} & \dots & C_{j_t, m_t} \end{bmatrix}$$

is invertible. However,

$$\begin{aligned}
 & \begin{bmatrix} C_{j_1, m_1} & \cdots & C_{j_1, m_t} \\ \vdots & \ddots & \vdots \\ C_{j_t, m_1} & \cdots & C_{j_t, m_t} \end{bmatrix} = \\
 & \begin{bmatrix} A_{r,k} A_{j_1,k}^{-1} A_{j_1, m_1} A_{r, m_1}^{-1} & \cdots & A_{r,k} A_{j_1,k}^{-1} A_{j_1, m_t} A_{r, m_t}^{-1} \\ \vdots & \ddots & \vdots \\ A_{r,k} A_{j_t,k}^{-1} A_{j_t, m_1} A_{r, m_1}^{-1} & \cdots & A_{r,k} A_{j_t,k}^{-1} A_{j_t, m_t} A_{r, m_t}^{-1} \end{bmatrix} = \\
 & \begin{bmatrix} A_{r,k} A_{j_1,k}^{-1} & & & \\ & \ddots & & \\ & & A_{r,k} A_{j_t,k}^{-1} & \\ & & & A_{r, m_1}^{-1} \\ & & & \ddots & \\ & & & & A_{r, m_t}^{-1} \end{bmatrix} \begin{bmatrix} A_{j_1, m_1} & \cdots & A_{j_1, m_t} \\ \vdots & \ddots & \vdots \\ A_{j_t, m_1} & \cdots & A_{j_t, m_t} \end{bmatrix}. \tag{36}
 \end{aligned}$$

Since each encoding matrix  $A_{i,j}$  is invertible, the first and the third matrices in (36) are invertible. The middle matrix is invertible since the code in (28) is invertible, and the result follows. ■

#### REFERENCES

- [1] M. Blaum, J. Brady, J. Bruck, and J. Menon, "EVENODD: an efficient scheme for tolerating double disk failures in RAID architectures," *IEEE Trans. on Comput.*, vol. 44, no. 2, pp. 192–202, Feb. 1995.
- [2] M. Blaum, J. Bruck, and E. Vardy, "MDS array codes with independent parity symbols," *IEEE Trans. on Inform. Theory*, vol. 42, no. 2, pp. 529–542, Mar. 1996.
- [3] V. R. Cadambe, C. Huang, and J. Li, "Permutation code: optimal exact-repair of a single failed node in MDS code based distributed storage systems," *Information Theory Proceedings (ISIT), IEEE International Symposium on*, pp. 1225 – 1229, Aug. 2011.
- [4] V. R. Cadambe, C. Huang, J. Li, and S. Mehrotra, "Polynomial length MDS codes with optimal repair in distributed storage," in *Signals, Systems and Computers (ASILOMAR), 2011 Conference Record of the Forty Fifth Asilomar Conference on*, Nov. 2011.
- [5] V. R. Cadambe, S. A. Jafar, and H. Maleki, "Minimum repair bandwidth for exact regeneration in distributed storage," *Wireless Network Coding Conference (WiNC), 2010 IEEE*, 2010.
- [6] V. R. Cadambe, S. A. Jafar, H. Maleki, K. Ramchandran, and C. Suh, "Asymptotic interference alignment for optimal repair of MDS codes in distributed data storage," [http://newport.eecs.uci.edu/~syed/papers/storage\\_final.pdf](http://newport.eecs.uci.edu/~syed/papers/storage_final.pdf), 2011.
- [7] P. Corbett, B. English, A. Goel, T. Grcanac, S. Kleiman, J. Leong, and S. Sankar, "Row-diagonal parity for double disk failure correction," *Proc. of the 3rd USENIX Symposium on File and Storage Technologies (FAST 04)*, 2004.
- [8] A. G. Dimakis, P. B. Godfrey, Y. Wu, M. J. Wainwright, and K. Ramchandran, "Network coding for distributed storage systems," *IEEE Trans. on Inform. Theory*, vol. 56, no. 9, pp. 4539–4551, Sep. 2010.
- [9] C. Huang and L. Xu, "STAR: An efficient coding scheme for correcting triple storage node failures," *IEEE Trans. on Comput.*, vol. 57, no. 7, pp. 889–901, Jul. 2008.
- [10] D. S. Papailiopoulos and A. G. Dimakis, "Distributed storage codes through Hadamard designs," in *Information Theory Proceedings (ISIT), IEEE International Symposium on*, Aug. 2011.
- [11] D. S. Papailiopoulos, A. G. Dimakis, and V. R. Cadambe, "Repair optimal erasure codes through hadamard designs," in *Communication, Control, and Computing (Allerton), 2011 49th Annual Allerton Conference on*, Sep. 2011.
- [12] K. V. Rashmi, N. B. Shah, and P. V. Kumar, "Enabling node repair in any erasure code for distributed storage," *Information Theory Proceedings (ISIT), IEEE International Symposium on*, pp. 1235 – 1239, Aug. 2011.
- [13] K. V. Rashmi, N. B. Shah, P. V. Kumar, and K. Ramchandran, "Explicit construction of optimal exact regenerating codes for distributed storage," *Allerton Conference on Control, Computing, and Communication, Urbana-Champaign, IL*, pp. 1243–1249, 2009.
- [14] N. Shah, K. Rashmi, P. 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, Apr. 2012.
- [15] N. Shah, K. 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, Mar. 2012.
- [16] C. Suh and K. Ramchandran, "Exact-repair MDS codes for distributed storage using interference alignment," *Information Theory Proceedings (ISIT), IEEE International Symposium on*, pp. 161–165, Jun. 2011.
- [17] I. Tamo, Z. Wang, and J. Bruck, "MDS array codes with optimal rebuilding," *Information Theory Proceedings (ISIT), IEEE International Symposium on*, pp. 1240–1244, Aug. 2011.
- [18] I. Tamo, Z. Wang, and J. Bruck, "Zigzag codes: MDS array codes with optimal rebuilding," *IEEE Trans. on Inform. Theory*, vol. 59, no. 3, pp. 1597–1616, Mar. 2013.
- [19] Z. Wang, I. Tamo, and J. Bruck, "On codes for optimal rebuilding access," in *Communication, Control, and Computing (Allerton), 2011 49th Annual Allerton Conference on*, Sep. 2011.
- [20] Z. Wang, I. Tamo, and J. Bruck, "Long MDS codes for optimal repair bandwidth," in *Information Theory Proceedings (ISIT), 2012 IEEE International Symposium on*, July 2012.
- [21] Y. Wu, "Existence and construction of capacity-achieving network codes for distributed storage," *Information Theory Proceedings (ISIT), IEEE International Symposium on*, pp. 1150 – 1154, 2009.
- [22] Y. Wu and A. G. Dimakis, "Reducing repair traffic for erasure coding-based storage via interference alignment," *Information Theory Proceedings (ISIT), IEEE International Symposium on*, pp. 2276 – 2280, 2009.
- [23] Y. Wu, A. G. Dimakis, and K. Ramchandran, "Deterministic regenerating codes for distributed storage," *Allerton Conference on Control, Computing, and Communication, Urbana-Champaign, IL*, 2007.

- [24] L. Xu, V. Bohossian, J. Bruck, and D. Wagner, "Low-density MDS codes and factors of complete graphs," *IEEE Trans. on Inform. Theory*, vol. 45, no. 6, pp. 1817–1826, Sep. 1999.
- [25] L. Xu and J. Bruck, "X-code: MDS array codes with optimal encoding," *IEEE Trans. on Inform. Theory*, vol. 45, no. 1, pp. 272–276, Jan. 1999.