Linear Exact Repair Rate Region of $(k+1,k,k)$ Distributed Storage Systems: A New Approach

Mehran Elyasi
Department of ECE
University of Minnesota
melyasi@umn.edu

Soheil Mohajer
Department of ECE
University of Minnesota
soheil@umn.edu

Ravi Tandon
Department of Computer Science
Virginia Tech
tandonr@vt.edu

***Abstract*—Characterizing the exact repair storage-vs-repair bandwidth tradeoff for distributed storage systems remains an open problem for more than four storage nodes. Motivated by the prevalence and practical applicability of linear codes, the exact repair problem when restricted to linear codes is considered. The main result of this paper is a new approach to develop bounds for exact repair distributed storage systems with linear codes (LDSS). Using this approach, the exact repair region for the $(k+1,k,k)$ LDSS is completely characterized. The new approach utilizes the properties of linear codes together with the exact repair constraints. These constraints are formally captured through an optimization problem with a recursive structure, and its solution finally yields the new bounds for the LDSS. These bounds together with recent code constructions characterize the exact repair region for $(k+1,k,k)$ LDSS.**

I. Introduction

Distributed storage systems (DSS) are increasingly being employed by various technologies. While the size of data, number of storage components, and number of users connecting to these servers are dramatically growing, efficiency of the system, in the sense of a fundamental tradeoff between the overhead penalty paid to provide robustness and the cost of system maintenance, is becoming the key factor to determine their performance. In order to reduce the computational complexity of encoding/decoding as well as system maintenance, *linearity* and *exact-repairability* are two essentially important properties required for practical purposes. Characterizing the optimal tradeoff between storage and repair-bandwidth is a challenging open problem for a general (n,k,d) DSS with exact repair property. Since the work of Dimakis et.al [1], which established the functional repair tradeoff, there have been several works on exact repair code constructions such as [4], [8]–[10] (also see the references therein), and non-achievability of some points on the functional repair tradeoff [3]. However, the fundamental question regarding the gap between functional and exact repair tradeoff remained open until recently shown by Tian in [5] (also see [7], [11] for more recent results).

In particular, for the $(4,3,3)$ system, it was shown in [5] that functional and exact repair tradeoffs are different through a novel *computer-aided* proof. Despite its originality, the solution involved an optimization problem over a large number of variables, making this approach non-scalable and intractable for larger system parameters. Moreover, such a computer-aided approach does not necessarily lead to intuition and insights which are very valuable for system design.

In practice, most of the codes that are currently being used in DSS implementations are linear codes (such as codes from the RAID family, and variations of Reed-Solomon codes). Motivated by the popularity and practical relevance of linear codes, in this paper, we focus on linear codes with the exact repair property. Henceforth, we refer *LDSS as distributed storage systems which employ linear codes*.

Our key approach is to use the underlying algebraic structure of linear codes to provide bounds on the performance of LDSS. In particular, we exploit the duality between the multi-dimensional subspaces over finite fields and linear codes in order to understand the structure of an optimal exact repair code. This results in a computationally feasible optimization problem to bound the performance of the system. Together with the recent code constructions in [9], our bounds characterize the exact repair region for the $(k+1,k,k)$ LDSS.

II. Problem Statement

We first describe the exact repair problem for distributed storage systems. An exact repair DSS with parameters $(n,k,d)=(k+1,k,k)$ and (α,β) consists of $n=k+1$ storage devices, each with storage capacity α. The entire data (Data) is encoded and stored distributedly over n nodes.

Notation: *We use $[i:j]=\{i,i+1,\dots,j\}$ to denote the set of positive integers between (and including) i and j. If $i=1$, we drop it, and simply use $[j]$ to denote set $\{1,2,\dots,j\}$, hence $[n]=\{1,2,\dots,n\}$ denotes the set of all node indices. We use W_i to denote the content stored in node i, and extend this definition to $W_A=\{W_i; i\in A\}$ for any $A\subseteq[n]$.*

In the $(k+1,k,k)$ DSS, the repair data sent from node i to repair node j is denoted by $\mathcal{S}_{i\to j}$. Note that since $n=d+1$, there is a unique way of choosing $d=k$ helper nodes to repair any failed node within $[n]$. Therefore, the dependence of $\mathcal{S}_{i\to j}$ on the remaining $(d-1)$ helper nodes, that is $[n]\setminus\{i,j\}$, is clear due to their uniqueness and is hence dropped from the notation for simplicity. We also set $\mathcal{S}_{i\to i}$, to be a dummy variable with zero entropy, for consistency. Moreover, $\mathcal{S}_{A\to B}=\{\mathcal{S}_{i\to j}: i\in A, j\in B\}$ denotes the total data sent by nodes in set A for the repair of nodes in set B.

The *data recovery* and *failure node repairability* are defined as follows.

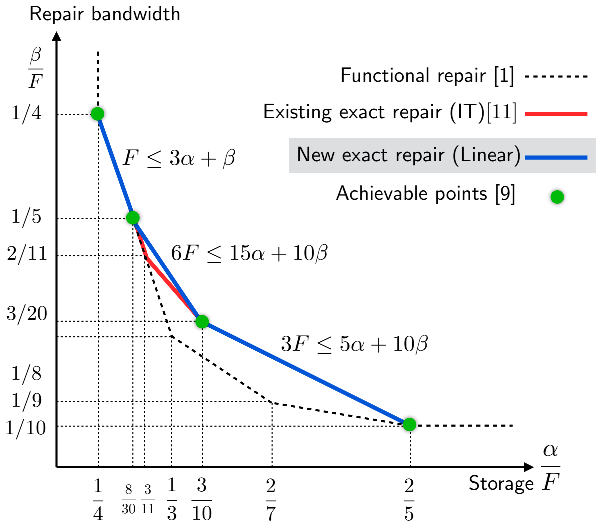

Fig. 1. Existing and new results for $(5, 4, 4)$ DSS.

- **MDS Property (data recovery):** The entire file can be recovered from the content of any k nodes: $H(\mathsf{Data}|W_A) = 0$ for any $A \subseteq [k+1]$ satisfying $|A| \geq k$.
- **Failure Node (Exact) Repairability:** The content of any failed node can be exactly recovered (repaired) by receiving no more that β units of repair data from any other d nodes, that is, $H(W_i|\mathcal{S}_{A\to i}) = 0$ for any $A \subseteq [k+1] \setminus \{i\}$, with $|A| \geq d$, where $H(\mathcal{S}_{i\to j}) \leq \beta$ and $H(\mathcal{S}_{i\to j}|W_i) = 0$.

The following theorem is the main result of this work which provides a new outer bound for the linear capacity of distributed storage systems with exact repair.

Theorem 1. *The exact repair capacity of a $(k+1, k, k)$ DSS with per node storage α and total node repair bandwidth $k\beta$ is upper bounded by*

$$F_k(\alpha, \beta) \leq \max_{(\pi_1,\pi_2,\dots,\pi_k)\in P_k(\alpha,\beta)} \Phi_k(\alpha, \beta; \pi_1, \pi_2, \dots, \pi_k). \tag{1}$$

where the maximum is taken over all feasible $(\pi_1, \pi_2, \dots, \pi_k)$, defined as

$$P_k(\alpha, \beta) = \left\{ \begin{array}{l} (\pi_1, \pi_2, \dots, \pi_k) : \\ 0 \leq \pi_1 \leq \pi_2 \leq \dots \leq \pi_k = \alpha \\ \pi_{k-1} \geq \alpha - \beta \\ \pi_i \geq \sum_{j=1}^{i} (-1)^{i-j} \binom{i}{j} \pi_j \end{array} \right\}. \tag{2}$$

The following corollary is direct consequences of Theorem 1, by solving the optimization problem[1] for $(5, 4, 4)$-DSS. Essentially, the outer bound given in Theorem 1 together with the code construction from [9] provide the complete characterization of the optimum tradeoff.

Corollary 1. *The optimum storage-bandwith exact-repair tradeoff of the $(5, 4, 4)$-DSS with linear codes is given by*

$$F_4(\alpha, \beta) \leq \min\left(4\alpha, 3\alpha + \beta, \frac{5}{2}\alpha + \frac{5}{3}\beta, \frac{5}{3}\alpha + \frac{10}{3}\beta, 10\beta\right).$$

III. Linear Distributed Storage System

In a linear code for DSS, the original file of size F is divided into F sub-packets. These sub-packets form a basis for an F-dimensional vector space, namely $\mathcal{F}$, over a finite filed $\mathbb{F}_q$, which determines the original file. Therefore, each unit of stored data can be expressed as a linear combination of the sub-packets, which is indeed a vector in $\mathcal{F}$, and thus, the content of each disk is equivalent to a subspace of $\mathcal{F}$. We denote by $\mathcal{W}_i$ the subspace spanned by the vectors stored on the i-th node. Moreover, $\mathcal{S}_{i\to j}$ denotes the subspace spanned by the vectors sent from node i to node j in order to repair j. In the following, we explain our approach to analyze the relationship between these subspaces with the exact repair property, and derive bounds on the optimum tradeoff of the LDSS.

In this new framework, the concept of exact regeneration code can be redefined as the following.

Definition 1. *An exact regeneration code for a $(k+1, k, k)$ DSS with parameters (α, β) is defined as a vector space $\mathcal{F}$ defined over a finite field $\mathbb{F}_q$, where*

i) *each node i stores a subspace $\mathcal{W}_i \subseteq \mathcal{F}$ with $\mathsf{dim}(\mathcal{W}_i) \leq \alpha$;*

ii) *the summation of every k node subspaces suffice to cover the entire vector space, that is, $\mathcal{F} = \sum_{i\in A} \mathcal{W}_i$ for every $A \subset [k+1]$ with $|A| = k$.*

iii) *in case of failure of node j, all other nodes can send a (at most) β-dimensional subspace of their own vector space, such that the missing vector space $\mathcal{W}_j$ can be recovered, that is $\mathcal{W}_j \subset \sum_{i\in[k+1]\setminus\{j\}} \mathcal{S}_{i\to j}$, where $\mathcal{S}_{i\to j} \subseteq \mathcal{W}_i$ and $\mathsf{dim}(\mathcal{S}_{i\to j}) \leq \beta$.*

Definition 2. *An achievable operation point (α, β) for an (n, k, d) DSS is called* pareto-optimum *if any other achievable point (α', β') satisfies either $\alpha' > \alpha$ or $\beta' > \beta$.*

Our focus is on the storage-repair bandwidth tradeoff of the optimum regeneration codes. In particular, in an optimum code we have $\mathsf{dim}(\mathcal{W}_i) = \alpha$ and $\mathsf{dim}(\mathcal{S}_{i\to j}) = \beta$, because otherwise, we can shrink at least one of the subspaces, and obtain a code with a better tradeoff.

We start with some basic definitions from linear algebra.

Definition 3. *We define the following operations between subspaces.*

i) *For two subspace $\mathcal{V}, \mathcal{U} \subseteq \mathcal{F}$ over $\mathbb{F}$, their intersection is defined as*

$$\mathcal{V} \cap \mathcal{U} = \{v : v \in \mathcal{V}, v \in \mathcal{U}\}.$$

The intersection of any two vector spaces is always a vector spaces.

[1] It is easy to show that the optimum bound of Tian [5] for $(4, 3, 3)$ can be also subsumed from this theorem.

ii) *For two subspace $\mathcal{V}$,$\mathcal{U} \subseteq \mathcal{F}$ over $\mathbb{F}$, their summation is defined as*

$$\mathcal{V} + \mathcal{U} = \{v + u : v \in \mathcal{V}, u \in \mathcal{U}\}.$$

The summation of two subspaces is called direct sum *and denoted by $\mathcal{V} \oplus \mathcal{U}$, if $\mathcal{U} \cap \mathcal{V} = \{\mathbf{0}\}$. Note that the summation of any two subspaces is also a vector space.*

A very important distinction between vector spaces defined on $\mathbb{R}$ (or $\mathbb{C}$) and those defined on finite fields is due to the fundamentally different concept of orthogonality. In real field, two orthogonal are always disjoint (except their trivial intersection at $\{\mathbf{0}\}$), while in vector spaces defined over finite fields, a vector can be orthogonal to it self (e.g. $[1,1] \cdot [1,1]^T = 0$ in $\mathbb{F}_2$). Hence, we need an alternative approach to define the complement of a subspace. This is formally defined as follows.

Definition 4. *Consider two vector spaces $\mathcal{U}$ and $\mathcal{V}$, and assume $\mathscr{B}(\mathcal{U}\cap\mathcal{V}) = \{v_1, \dots, v_p\}$ forms a linearly independent basis for $\mathcal{U} \cap \mathcal{V}$. We can extend this set of vectors, to $\mathscr{B} = \{v_1, v_2, \dots, v_p, v_{p+1}, \dots, v_q\}$, to form a basis for $\mathcal{U}$. Then we define $[\mathcal{U}]_{\mathsf{mod}(\mathcal{V})}$ ($\mathcal{U}$* after nulling *$\mathcal{V}$) as*

$$[\mathcal{U}]_{\mathsf{mod}(\mathcal{V})} \triangleq \mathsf{span}(\mathscr{B}(\mathcal{U}) \backslash \mathscr{B}(\mathcal{U}\cap\mathcal{V})) = \mathsf{span}(\{v_{p+1}, \dots, v_q\}).$$

In other words, one can think of $[\mathcal{U}]_{\mathsf{mod}(\mathcal{V})}$ as the complement of $\mathcal{U}$ with respect to $\mathcal{V}$, that is $\mathcal{U} \subseteq [\mathcal{U}]_{\mathsf{mod}(\mathcal{V})} \oplus \mathcal{V}$.

The following lemma characterizes some of the basic properties of this operator. The proof of this lemma is just based on basic concepts of linear algebra, and hence skipped here for sake of brevity.

Lemma 1. *For any two vector spaces $\mathcal{A}$ and $\mathcal{B}$, we have*

$P1)$ $[\cdot]_{\mathsf{mod}(\mathcal{U})}$ *is a linear operator on vector spaces, that is,*

$$[\mathcal{U}_1 + \mathcal{U}_2]_{\mathsf{mod}(\mathcal{V})} = [\mathcal{U}_1]_{\mathsf{mod}(\mathcal{V})} + [\mathcal{U}_2]_{\mathsf{mod}(\mathcal{V})}.$$

$P2)$ *More generally, $[\cdot]_{\mathsf{mod}(\mathcal{U})}$ preserves subspace relationship, i.e., if $\mathcal{U}_1 \subseteq \mathcal{U}_2$, then $[\mathcal{U}_1]_{\mathsf{mod}(\mathcal{V})} \subseteq [\mathcal{U}_2]_{\mathsf{mod}(\mathcal{V})}$.*

$P3)$ $\mathsf{dim}([\mathcal{U}]_{\mathsf{mod}(\mathcal{V})}) = \mathsf{dim}(\mathcal{U}) - \mathsf{dim}(\mathcal{U} \cap \mathcal{V})$.

Definition 5. *Let $A \subset [k+1]$ be a subset of nodes and node $i \notin A$. Consider the vector space formed by the sum of the subspaces of the nodes in A. We denote the intersection of this subspace and $\mathcal{W}_i$ by*

$$\mathcal{W}(A;i) \triangleq \left(\sum\nolimits_{j\in A} \mathcal{W}_j\right) \cap \mathcal{W}_i.$$

Definition 6. *Consider a symmetric $(k+1,k,k)$ DSS with a regeneration code operating at parameters (α, β). We call the code a $(\pi_1, \pi_2, \dots, \pi_k)$-code if the associated vector spaces to the nodes satisfy*

$$\mathsf{dim}\left(\left(\sum\nolimits_{j\in A} \mathcal{W}_j\right) \cap \mathcal{W}_i\right) = \pi_{|A|} \tag{3}$$

for every $A \subset [n]$ and $i \notin A$.

A sequence $\underline{\pi} = (\pi_1, \pi_2, \dots, \pi_k)$ is called feasible *if there exists a $(\pi_1, \pi_2, \dots, \pi_k)$ regeneration code for $(k+1,k,k)$ DSS.*

Note that elements of $\underline{\pi}$ are jointly defined over a family of subspaces. Hence, they are not isolated (independent of each other), and any sequence $\underline{\pi}$ associated with a regeneration code should satisfy a set of constraints.

The following lemma states some necessary conditions on feasible sequences.

Lemma 2. *A sequence $(\pi_1, \pi_2, \dots, \pi_k)$ is feasible to be associated with a family of subspaces $(\mathcal{W}_1, \mathcal{W}_2, \dots, \mathcal{W}_{k+1})$, only if*

(1) $0 \le \pi_1 \le \pi_2 \le \cdots \le \pi_k = \alpha$.

(2) $\pi_{k-1} \ge \alpha - \beta$

(3) $\pi_i \ge \sum_{j=1}^{i} (-1)^{i-j} \binom{i}{j} \pi_j$.

We denote the set of all $\underline{\pi}$-vectors satisfying (1), (2), (3) by $P(\alpha, \beta)$.

Proof of Lemma 2. First note that in any $(\pi_1, \pi_2, \dots, \pi_k)$-code, all the π_i's are non-negative integers. Moreover, for any pair of sets (A, B) with $B = A \cup \{j\}$ (where $j \notin A$), we have $\sum_{i\in A} \mathcal{W}_i \subset (\sum_{i\in A} \mathcal{W}_i) + \mathcal{W}_j = \sum_{i\in B} \mathcal{W}_i$. Hence,

$$\mathcal{W}(A;i) \subset \mathcal{W}(B;i)$$

which implies

$$\pi_{|A|} = \mathsf{dim}(\mathcal{W}(A;i)) \le \mathsf{dim}(\mathcal{W}(B;i)) = \pi_{|A|+1},$$

which shows $\underline{\pi}$ is a non-decreasing sequence.

Next, note that for any $(k+1,k,k)$ DSS, the entire data can be recovered from the first k nodes. That means $\mathcal{W}_{k+1} \subset \mathcal{F} = \sum_{i\in[k]} \mathcal{W}_i$, which implies

$$\begin{aligned}\pi_k &= \mathsf{dim}\left(\left(\sum\nolimits_{i\in[k]} \mathcal{W}_i\right) \cap \mathcal{W}_{k+1}\right) \\ &= \mathsf{dim}(\mathcal{F} \cap \mathcal{W}_{k+1}) = \mathsf{dim}(\mathcal{W}_{k+1}) = \alpha.\end{aligned}$$

This completes the proof of (1).

In order to show (2), recall that $\mathcal{W}_{k+1}$ can be reconstructed by the sum of the repair data $\mathcal{S}_{i\to k+1}$ sent from node $i = 1, 2, \dots, k$ to $k+1$, i.e., $\mathcal{W}_{k+1} \subseteq \sum_{i=1}^{k} \mathcal{S}_{i\to k+1}$. Hence, since $\mathcal{S}_{i\to k+1} \subseteq \mathcal{W}_i$, we have

$$\begin{aligned}\alpha = \mathsf{dim}(\mathcal{W}_k) &= \mathsf{dim}\left(\left(\sum\nolimits_{i=1}^{k} \mathcal{S}_{i\to k+1}\right) \cap \mathcal{W}_{k+1}\right) \\ &\le \mathsf{dim}\left(\left(\mathcal{S}_{1\to k+1} + \sum\nolimits_{i=2}^{k} \mathcal{W}_i\right) \cap \mathcal{W}_{k+1}\right) \\ &\le \mathsf{dim}\left(\left(\sum\nolimits_{i=2}^{k} \mathcal{W}_i\right) \cap \mathcal{W}_{k+1}\right) + \mathsf{dim}(\mathcal{S}_{1\to k+1}) \\ &= \pi_{k-1} + \beta.\end{aligned}$$

This implies $\pi_{k-1} \ge \alpha - \beta$.

Finally, last constraint is based on inclusionexclusion principle. The complete proof is rather involved and refer the interested reader to our technical report [12]. However, the sketch of the proof is as follows. Consider a set $A \subset [k+1]$ with $|A| = i$ and $\ell \notin A$. It is clear that

$$\mathcal{W}(B, \ell) \subseteq \mathcal{W}(A, \ell), \qquad \forall B \subseteq A.$$

Now, consider all $\binom{i}{i-1}$ subsets of A each of size $(i-1)$. Each of them contribute a π_{i-1}-dimensional subspace into $\mathcal{W}(A,\ell)$. However, by summing up all these dimensions, we are over counting their intersection. Let B, C be two subsets of A, each of size $(i-1)$ such that $|D| = |B \cap C| = i-2$. Since $\mathcal{W}(D,\ell) \subseteq \mathcal{W}(B,\ell)$ and $\mathcal{W}(D,\ell) \subseteq \mathcal{W}(C,\ell)$, every vector in $\mathcal{W}(D,\ell)$ is over counted in $\binom{i}{i-1}\pi_{i-1}$. Thus, we compensate for that by subtracting all those over-counting by subtracting $\binom{i}{i-2}\pi_{i-2}$. A similar argument can be followed to obtain the desired inequality. □

Next, we define the linear capacity of a distributed storage system that employs a $(\pi_1, \pi_2, \dots, \pi_k)$-code.

Definition 7. *We denote the capacity of a $(k+1,k,k)$ DSS operating at (α,β) and using a $(\pi_1,\pi_2,\dots,\pi_k)$-code by $\Phi_k(\pi_1,\pi_2,\dots,\pi_k)$.*

Recall that symmetric regeneration codes achieve the optimum capacity of DSS. On the other hand, any symmetric code, is associated with some feasible $\underline{\pi} \in \mathbb{P}(\alpha,\beta)$. Hence, we can immediately conclude the following lemma.

Lemma 3. *The linear capacity of a $(n,k,d) = (k+1,k,k)$ DSS with per node capacity α and total repair bandwith $d\beta$ is given by*

$$F_k(\alpha,\beta) \le \max_{(\pi_1,\pi_2,\dots,\pi_k)\in P(\alpha,\beta)} \Phi_k(\alpha,\beta;\pi_1,\pi_2,\dots,\pi_k). \quad (4)$$

The following theorem allows us to construct an exact-repair regeneration code for a $(k,k-1,k-1)$ system from any existing symmetric code for a $(k+1,k,k)$ DSS.

Theorem 2. *Let $(\mathcal{W}_1,\mathcal{W}_2,\dots,\mathcal{W}_k,\mathcal{W}_{k+1})$ be a $(\pi_1,\pi_2,\dots,\pi_k)$-regeneration code for a $(k+1,k,k)$-DSS, that operates at (α,β). Then $(\mathcal{W}'_1,\mathcal{W}'_2,\dots,\mathcal{W}'_k)$ is a $(\pi_2-\pi_1,\pi_3-\pi_2,\dots,\pi_k-\pi_1)$-regeneration code for a $(k,k-1,k-1)$-DSS with parameters $(\alpha',\beta') = (\alpha-\pi_1,\beta)$, where*

$$\mathcal{W}'_i = [\mathcal{W}_i]_{\mathsf{mod}(\mathcal{W}_{k+1})}.$$

Moreover, this code can store $\mathcal{F}' = [\mathcal{F}]_{\mathsf{mod}(\mathcal{W}_{k+1})}$ on the system, where $\mathsf{dim}(\mathcal{F}') = \mathsf{dim}(\mathcal{F}) - \min(\alpha,k\beta)$.

This theorem is the core contribution of this work, and we present its proof in Section IV.

Using Theorem 2, we can obtain a recursive bound on the exact-repair capacity of any distributed storage system under linear codes.

The following lemma is a direct consequence of Theorem 2, which establishes a recursive upper bound on the $\Phi_k(\cdot)$ function defined above.

Lemma 4. *For any operation point (α,β), positive integer k, and any feasible sequence $\underline{\pi} \in P(\alpha,\beta)$, we have*

$$\begin{aligned}\Phi_k(\alpha,\beta;\pi_1,\pi_2,\dots,\pi_k) &\le \min(\alpha,k\beta)\\ &+ \Phi_{k-1}(\alpha-\pi_1,\beta;\pi_2-\pi_1,\pi_3-\pi_1,\dots,\pi_k-\pi_1) \quad (5)\end{aligned}$$

Proof of Lemma 4. Consider a $(k+1,k,k)$ DSS with parameters (α,β). Let $\underline{\pi} = (\pi_1,\dots,\pi_k)$ be a feasible sequence for this system. Consider an *optimum* code for this system, which can stores $\Phi_k(\alpha,\beta;\pi_1,\pi_2,\dots,\pi_k)$ units of data on the system.

From Theorem 2, we can null the content of the last node in this code $\mathcal{W}_{k+1}$, to obtain a new code for the resulting $(k,k-1,k-1)$ system with parameters $(\alpha',\beta') = (\alpha-\pi,\beta)$. Regardless of this new code being optimum for the new system or not, it can at most store $\Phi_{k-1}(\alpha-\pi_1,\beta;\pi_2-\pi_1,\pi_3-\pi_1,\dots,\pi_k-\pi_1)$ units of data. However, Theorem 2 implies by nulling we do not loose more that $\mathsf{dim}(\mathcal{W}_{k+1}) = \min(\alpha,k\beta)$ units of data on the original system. Hence, the total dimension (file size) of the original system could be at most $\min(\alpha,k\beta) + \Phi_{k-1}(\alpha-\pi_1,\beta;\pi_2-\pi_1,\pi_3-\pi_1,\dots,\pi_k-\pi_1)$. This completes the proof of this lemma.

□

IV. Proof of Theorem 2

Consider a $(k+1,k,k)$ DSS operating at (α,β) using a $(\pi_1,\pi_2,\dots,\pi_k)$-code. We denote by $\mathcal{W}_i$ the vector space of the i-the node. We null $\mathcal{W}_{k+1}$ in the entire system. Note that $\mathcal{W}'_{k+1} = \{\mathbf{0}\}$, since this operation maps every vector in $\mathcal{W}_{k+1}$ to the zero vector, and makes the last disk empty. Also the entire vector space remained in this system after nulling $\mathcal{W}_{k+1}$ is $\mathcal{F}' = [\mathcal{F}]_{\mathsf{mod}(\mathcal{W}_{k+1})}$, where

$$\mathsf{dim}(\mathcal{F}') \overset{(a)}{=} \mathsf{dim}(\mathcal{F}) - \mathsf{dim}(\mathcal{W}_{k+1}) = \mathsf{dim}(\mathcal{F}) - \min(\alpha,k\beta).$$

Note that in (a) we used Property $P3$, together with the fact that $\mathcal{W}_{k+1} \subseteq \mathcal{F}$ which leads to $\mathcal{F} \cap \mathcal{W}_{k+1} = \mathcal{W}_{k+1}$.

We prove the claims of the theorem through the following steps:

Data recovery: We need to show that $\mathcal{F}' = \sum_{i\in A}\mathcal{W}'_i$ for every $A \subseteq [k]$ with $|A| = k-1$. Consider such an A, and define $B = A \cup \{k+1\}$, so that $|B| = k$. Recall that data recovery property of the original system with $(k+1)$ nodes implies

$$\mathcal{F} = \sum_{i\in B}\mathcal{W}_i = \mathcal{W}_{k+1} + \sum_{i\in A}\mathcal{W}_i.$$

Hence, using linearity of $[\cdot]_{\mathsf{mod}(\mathcal{W}_{k+1})}$, we have

$$\begin{aligned}\mathcal{F}' = [\mathcal{F}]_{\mathsf{mod}(\mathcal{W}_{k+1})} &= \left[\mathcal{W}_{k+1} + \sum_{i\in A}\mathcal{W}_i\right]_{\mathsf{mod}(\mathcal{W}_{k+1})}\\ &\overset{(b)}{=} [\mathcal{W}_{k+1}]_{\mathsf{mod}(\mathcal{W}_{k+1})} + \sum_{i\in A}[\mathcal{W}_i]_{\mathsf{mod}(\mathcal{W}_{k+1})}\\ &= \{\mathbf{0}\} + \sum_{i\in A}\mathcal{W}'_i = \sum_{i\in A}\mathcal{W}'_i, \quad (6)\end{aligned}$$

where (b) follows from Property $P1$. This shows that nodes in any subset of size $(k-1)$ in the new system are able to recover the data. Also note that

$$\begin{aligned}\mathsf{dim}(\mathcal{W}_i) &= \mathsf{dim}([\mathcal{W}_i]_{\mathsf{mod}(\mathcal{W}_{k+1})})\\ &= \mathsf{dim}(\mathcal{W}_i) - \mathsf{dim}(\mathcal{W}_i \cap \mathcal{W}_{k+1}) = \alpha - \pi_1. \quad (7)\end{aligned}$$

Node Repairability: Similar to the above argument, we start with the repair of some node $j \in [k]$ in the original system. Since $(\mathcal{W}_1, \dots, \mathcal{W}_{k+1})$ is an exact-repair code, node i can be repaired by the help of other k nodes in $A = [k+1] \setminus \{j\}$, that is $\mathcal{W}_j \subseteq \sum_{i\in A} \mathcal{S}_{i\to j}$. Hence, since $[\cdot]_{\mathsf{mod}(\mathcal{W}_{k+1})}$ preserves subspace relationship, we have

$$\begin{aligned}
\mathcal{W}'_j = [\mathcal{W}_j]_{\mathsf{mod}(\mathcal{W}_{k+1})} &\overset{(c)}{\subseteq} \left[\sum_{i\in A} \mathcal{S}_{i\to j}\right]_{\mathsf{mod}(\mathcal{W}_{k+1})} \\
&= [\mathcal{S}_{k+1\to j}]_{\mathsf{mod}(\mathcal{W}_{k+1})} + \left[\sum_{i\in A\setminus\{k+1\}} \mathcal{S}_{i\to j}\right]_{\mathsf{mod}(\mathcal{W}_{k+1})} \\
&= \{\mathbf{0}\} + \sum_{i\in A\setminus\{k+1\}} [\mathcal{S}_{i\to j}]_{\mathsf{mod}(\mathcal{W}_{k+1})} \\
&= \sum_{i\in A\setminus\{k+1\}} \mathcal{S}'_{i\to j}, \qquad (8)
\end{aligned}$$

where in (c) we used Property $P2$. This implies any node j in the new system is repairable using $\mathcal{S}'_{i\to j}$ repair data from all other nodes. It is also worth mentioning that

$$\mathsf{dim}(\mathcal{S}'_{i\to j}) \le \mathsf{dim}(\mathcal{S}_{i\to j}) \le \beta.$$

So, we have a linear $(k, k-1, k-1)$ DSS with exact repair property, with parameters $(\alpha', \beta') = (\alpha - \pi_1, \beta)$.

It only remains to show that for the new code we have

$$\pi'_{|A|} = \mathsf{dim}\left(\left(\sum\nolimits_{j\in A} \mathcal{W}'_j\right) \cap \mathcal{W}'_i\right) = \pi_{|A|+1} - \pi_1,$$

for every pair of $A \subset [k]$ and $i \notin A$. Without loss of generality, we can consider $i = 1$ and $A \subset \{2, \dots, k\}$. Let $B = A \cup \{k+1\}$.

First note that $\mathcal{W}_{k+1} \cap \mathcal{W}_1 \subseteq \left(\sum_{j\in B} \mathcal{W}_j\right) \cap \mathcal{W}_1$. Assume $\mathscr{B}_1 = \{v_1, v_2, \dots, v_{\pi_1}\}$ be a basis vector for $\mathcal{W}_{k+1} \cup \mathcal{W}_1$. Extend it to $\mathscr{B}_2 = \{v_1, \dots, v_{\pi_1}, v_{\pi_1+1}, \dots, v_{\pi_{|B|}}\}$ to form a basis for the larger subspace $\left(\sum_{j\in B} \mathcal{W}_j\right) \cap \mathcal{W}_1$. These vectors will be mapped to

$$v'_\ell = [v_\ell]_{\mathsf{mod}(\mathcal{W}_{k+1})}, \qquad \ell = 1, \dots, \pi_{|B|}.$$

This is easy to see, by linearity of $[\cdot]_{\mathsf{mod}(\mathcal{W}_{k+1})}$, that every vector in $\left(\sum_{j\in A} \mathcal{W}'_1\right) \cap \mathcal{W}'_i$ can be written as a linear combination of elements in $\{v'_1, \dots, v'_{\pi_{|B|}}\}$. Recall that $v_\ell \in \mathcal{W}_{k+1}$ for $\ell = 1, \dots, \pi_1$, and hence, after nulling $\mathcal{W}_{k+1}$, we have $v'_\ell = \mathbf{0}$ for $\ell = 1, \dots, \pi_1$.

On the other hand, we can show that elements of $\{v'_{\pi_1+1}, \dots, v'_{\pi_{|B|}+1}\}$ are linearly independent. To this end, assume

$$\sum_{\ell=\pi_1+1}^{\pi_{|B|}} \lambda_\ell v'_\ell = \mathbf{0},$$

for some set of coefficients $\lambda_{\pi_1+1}, \dots, \lambda_{\pi_{|B|}} \in \mathbb{F}_q$. Then, we have

$$\left[\sum_{\ell=\pi_1+1}^{\pi_{|B|}} \lambda_\ell v_\ell\right]_{\mathsf{mod}(\mathcal{W}_{k+1})} = \sum_{\ell=\pi_1+1}^{\pi_{|B|}} \lambda_\ell v'_\ell = \mathbf{0},$$

which further implies $\sum_{\ell=\pi_1+1}^{\pi_{|B|}} \lambda_\ell v_\ell \in \mathcal{W}_{k+1}$, because it is a vector which is mapped to $\mathbf{0}$ by nulling $\mathcal{W}_{k+1}$. On the other hand we know, $\sum_{\ell=\pi_1+1}^{\pi_{|B|}} \lambda_\ell v_\ell \in \mathcal{W}_1$, since each of v_ℓ's lie in $\mathcal{W}_1$. This implies $\sum_{\ell=\pi_1+1}^{\pi_{|B|}} \lambda_\ell v_\ell \in \mathcal{W}_{k+1} \cap \mathcal{W}_1 = \mathsf{span}(\{v_1, \dots, v_{\pi_1}\})$. Hence, there exists a set of coefficients $\delta_1, \dots, \delta_{\pi_1}$ such that

$$\sum_{\ell=\pi_1+1}^{\pi_{|B|}} \lambda_\ell v_\ell = \sum_{\ell=1}^{\pi_1} \delta_\ell v_\ell.$$

This is only possible if all λ_ℓ's and δ_ℓ's are zero, since $\mathscr{B}_2 = \{v_\ell : \ell = 1, ;\pi_{|B|}\}$ was a basis for $\left(\sum_{j\in B} \mathcal{W}_j\right) \cap \mathcal{W}_1$, and thus consists of linearly independent vectors.

From this argument we can conclude that $\mathscr{B}_2 \subseteq \mathscr{B}_1$ forms a basis for $\left(\sum_{j\in A} \mathcal{W}'_j\right) \cap \mathcal{W}'_1$, and hence

$$\begin{aligned}
\pi_{|A|} &= \mathsf{dim}\left(\left(\sum_{j\in A} \mathcal{W}'_j\right) \cap \mathcal{W}'_1\right) \qquad (9) \\
&= |\mathscr{B}_2 \subseteq \mathscr{B}_1| = \pi_{|A|+1} - \pi_1. \qquad (10)
\end{aligned}$$

This completes the proof of Theorem 2.

References

[1] A. G. Dimakis, P. B. 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, Sept. 2010.

[2] R. Ahlswede, N. Cai, S.-Y. R. Li and R. W. Yeung, "Network information flow," *IEEE Trans. Inf. Theory,* vol. 46, pp. 1204–1216, Jul. 2000.

[3] N. B. Shah, K. V. Rashmi, P. V. Kumar and K. Ramchandran, "Distributed storage codes with repair-by-transfer and non-achievability of interior points on the storage-bandwidth tradeoff," *IEEE Trans. Inf. Theory,* vol. 58, no. 3, pp. 1837–1852, Mar. 2012.

[4] V. Cadambe, S. Jafar, H. Maleki, K. Ramchandran and C. Suh, "Asymptotic interference alignment for optimal repair of MDS codes in distributed storage," *IEEE Trans Inf. Theory,* vol. 59, no. 5, pp. 2974–2987, May. 2013.

[5] C. Tian, "Rate region of the (4,3,3) exact-repair regenerating codes," in *Proc. Intern. Symp. Inf Theory, ISIT*, Istanbul, Turkey, Jun. 2013.

[6] T. M. Cover and J. A. Thomas, "Elements of Information Theory", 1991. New York: Wiley.

[7] B. Sasidharan, K. Senthoor, P. V. Kumar , "An Improved Outer Bound on the Storage-Repair-Bandwidth Tradeoff of Exact-Repair Regenerating Codes", in arXiv:1312.6079, Dec. 2013.

[8] C. Tian, B. Sasidharan, V. Aggarwal, V. A. Vaishampayan, P. V. Kumar, "Layered, Exact-Repair Regenerating Codes Via Embedded Error Correction and Block Designs", in arXiv:1408.0377, Aug. 2014.

[9] S. Goparaju, S. El Rouayheb, R. Calderbank , "New Codes and Inner Bounds for Exact Repair in Distributed Storage Systems", in arXiv:1402.2343, Feb. 2014.

[10] K. V. Rashmi, N. B. Shah, and P. V. Kumar, "Optimal Exact-Regenerating Codes for Distributed Storage at the MSR and MBR Points via a Product-Matrix Construction," in *IEEE Transactions on Information Theory*, vol. 57, pp. 5227–5239, Aug. 2011.

[11] S. Mohajer and R. Tandon, "Exact Repair for Distributed Storage Systems: Partial Characterization via New Bounds", Information Theory and Applications Workshop (ITA), San Diego, Feb. 2015.

[12] S. Mohajer and R. Tandon, "Exact Repair for Linear Distributed Storage Systems", *Technical Report.*, Available at http://www.ece.umn.edu/~soheil/Publications_files/Tech_Rep_1.pdf