Explicit Constructions of MSR Codes for Clustered Distributed Storage: The Rack-Aware Storage Model

Zitan Chen and Alexander Barg, *Fellow, IEEE*

***Abstract*— The paper is devoted to the problem of erasure coding in distributed storage. We consider a model of storage that assumes that nodes are organized into equally sized groups, called racks, that within each group the nodes can communicate freely without taxing the system bandwidth, and that the only information transmission that counts is the one between the racks. This assumption implies that the nodes within each of the racks can collaborate before providing information to the failed node. The main emphasis of the paper is on code construction for this storage model. We present an explicit family of maximum distance separable (MDS) array codes that support recovery of a single failed node from any number of helper racks using the minimum possible amount of inter-rack communication(such codes are said to provide optimal repair). The codes are constructed over finite fields of size comparable to the code length. We also derive a bound on the number of symbols accessed at helper nodes for the purposes of repair, and construct a code family that approaches this bound, while still maintaining the optimal repair property. Finally, we present a construction of scalar Reed-Solomon codes that support optimal repair for the rack-oriented storage model.**

***Index Terms*— Distributed storage, clustered storage, MSR codes, multiplicative group, optimal repair.**

I. Introduction

ERASURE codes increase reliability and efficiency of distributed storage by supporting the recovery of the data on failed nodes under the restriction of low repair bandwidth, i.e., limited amount of information downloaded from other nodes for the purposes of the repair. This problem was initially introduced in the well-known paper [5] which cast the capacity problem of distributed storage as a network coding problem where the necessary conditions for the repair of failed nodes were derived by considering the information flow in the network that occurred in the course of repair. These conditions imply a bound on the minimum number of symbols required for repair of a single failed node, which is known as the cut-set bound on the repair bandwidth. Paper [5] further considered a variety of data coding schemes that optimize either storage or repair bandwidth, as well as the tradeoff between these two quantities. In this paper we limit ourselves to minimum-storage regenerating (MSR) codes, or, equivalently, to MDS codes with optimal repair. We further restrict our attention to the task of exact repair as opposed to more general functional repair [5].

Initially the repair problem was formulated for the so-called centralized repair model which assumes that the failed nodes are repaired by a single data collector that receives information from the helper nodes and performs the recovery within a single location, having full access to all the downloaded information and the intermediate results of the calculations [4], [17], [28]. Another well-known model assumes cooperative repair, when the failed nodes are restored at different physical locations, and the information downloaded to each of them as well as the exchange of intermediate results between them are counted toward the overall repair bandwidth [11], [12], [20], [29].

The problems of centralized and cooperative repair have been addressed in multiple recent papers, and there are explicit constructions of optimal-repair regenerating codes that cover the entire range of admissible parameters, require small-size ground alphabet compared to the length n of the encoding block, and attain the smallest possible repair bandwidth [16], [23], [28], [19], [27], [30], [13] (more references are given in a recent survey [2]). The availability of optimal constructions has motivated a shift of attention toward studying data recovery not only under communication, but also *connectivity constraints*, in other words, storage models in which communication cost between nodes differs depending on their location in the storage cluster. One of the simple extensions from the basic setting of homogeneous storage suggests that the nodes are joined into several groups (clusters), and repair of a node can be based on information from both the nodes within its own group and from nodes in the other groups. This permits to differentiate between communication within the cluster and the inter-cluster downloads, and the natural assumption is that the former is easier (contributes less to the repair bandwidth) than the latter.

Erasure coding for clustered architectures was introduced several years ago and affords several variations. One of the first questions analyzed for heterogeneous storage models was related to repair under the condition that the system contains

Manuscript received January 17, 2019; revised July 28, 2019; accepted September 10, 2019. Date of publication September 16, 2019; date of current version January 20, 2020. Z. Chen was supported by the NSF under Grant CCF1618603. A. Barg was supported by the NSF under Grant CCF1618603 and Grant CCF1814487. This article was presented in part at the Allerton Conference on Communication, Control and Computing, and at the 2019 IEEE International Symposium on Information Theory.

Z. Chen is with the Department of ECE and ISR, University of Maryland, College Park, MD 20742 USA (e-mail: chenztan@gmail.com).

A. Barg is with the Department of ECE and ISR, University of Maryland, College Park, MD 20742 USA, and also with IITP, Russian Academy of Sciences, 127051 Moscow, Russia (e-mail: abarg@umd.edu).

Communicated by A. Jiang, Associate Editor for Coding Theory.

Digital Object Identifier 10.1109/TIT.2019.2941744

a group of nodes, downloading information from which contributes more to the repair bandwidth than downloading the same amount of information from the other nodes [1]. Later works [6], [14] observed that a more realistic version of non-homogeneous storage should assume that the cost of downloading information depends on the relative location of the failed node in the system. In this case, downloading information from the group that contains the failed node (also called the *host group*) contributes less to the cost than inter-cluster downloads. The authors of [6], [14] have assumed that the storage is formed of two clusters and derived versions of the cut-set bound for the minimum repair bandwidth. The two-cluster model was further developed in recent papers [21], [22] which assumed that the encoded data is placed in a number of clusters (generally more than two), and derived a cut-set type bound on the repair bandwidth for this case. Moreover, [22] showed existence of optimal-repair codes for their model, and [21] gave an explicit construction of MDS array codes for the case when the code dimension is equal to the size of the cluster. Note however that these results do not allow data processing within clusters in the course of the repair task, and thus are not directly comparable with our findings. Paper [15] considers several versions of data repair for clustered (rack-aware) storage, but does not address the general case of rack-aware MSR codes. We also mention [18], [26], [28] which discuss other variations of clustered storage architectures and are less related to our work.

A. Rack-Aware Storage

The model that we address in this work assumes that k data blocks are encoded into a codeword of length $n = \bar{n}u$ and stored across n nodes. The nodes are organized into $\bar{n}$ groups, also called racks. Suppose that a node has failed and call the rack that contains it the *host rack*. To perform the repair, the system downloads information from the nodes in the host rack (called below *local nodes*), as well as information from the other racks. The rack-oriented storage model is distinguished from the other clustered storage architectures in that the information from nodes that share the same rack, can be processed before communicating it to the failed node. Communication within the racks, including the host rack, does not incur any cost toward the repair bandwidth. The main benefit of rack-aware coding is related to reducing the bandwidth required for repair compared to coding for homogeneous storage.

This model was introduced in [9], [10]. Specifically, the authors of [9] derived a version of the cut-set bound of [5] adapted for this case and showed existence of minimum-storage codes with optimal repair for the rack model. A more expanded study of codes for this model, both for the minimum-storage and minimum-bandwidth scenarios, was undertaken in a recent paper [8], which showed existence of codes with optimal repair bandwidth for a wide range of parameters. At the same time, there are very few explicit constructions of MSR codes for this model known in the literature. We mention [10] which presented such codes for 3 racks and for the case when the number of parities of the code equals the size of the rack u.

B. Main Results

In this paper we present constructions of minimum-storage regenerating codes for the rack-aware storage model that have optimal repair bandwidth and cover all admissible parameters, such as the code rate k/n, the size and number of the racks. The only restriction that we assume is the natural condition that the racks are of equal size u and that the codeword is written on $\bar{n}$ racks such that u coordinates of the codeword are placed on each of them. This assumption is also consistent with the literature [8], [9].

We present two families of MDS array codes that support optimal repair in the rack model. The first family gives an explicit construction of optimal-bandwidth codes for repairing a *single node* from the nodes located in $\bar{d}$ helper racks for any $\lfloor k/u \rfloor \le \bar{d} \le \bar{n} - 1$. The underlying finite field of our construction is of size at most n^2/u where u is the size of the rack, and the node size (sub-packetization) equals $l \approx (\bar{d} - \frac{k}{u})^{n/u}$. The construction is phrased in terms of the parity-check equations of the code, as in [28], [30], and relies on the multiplicative structure of the field to account for the rack model considered here.

The second code family constructed in this paper, in addition to optimal repair, addresses the question of reducing the number of symbols accessed on each of the helper racks. The code construction is presented in two steps. First, we present a new family of optimal-access codes for the standard repair problem (homogeneous storage), constructing codes with arbitrary repair degree $d, k \le d \le n - 1$ over a field F of size at least $d - k + 1$. These parameters are similar to optimal-access codes constructed in [28], and in fact require a slightly larger field F. At the same time, the new construction can be modified for the rack model, resulting in codes with low access.

We also present a family of (scalar) Reed-Solomon codes that can be optimally repaired in the rack model. The construction is a modified version of the RS code family constructed in [25] for the case of homogeneous storage.

Apart from the code constructions, we examine the structure of codes with optimal repair or optimal access for the rack model. Because of intra-rack processing, the definition of optimal access is not as explicit as in the homogeneous case. We prove a lower bound on the number of accessed symbols for codes that support optimal repair. At the same time, the codes that we construct fall short of attaining this bound, and it is not clear what is the correct value of this quantity.

Finally, we derive a lower bound on the node size for optimal-repair codes in the rack model, modifying for this purpose the approach of the recent work [3], where a similar bound was proved for the homogeneous case.

II. Problem Statement and Structural Lemmas

Assume that the data file of size M is divided into k blocks and encoded using an array code $\mathcal{C}$ of length n over some finite field F. Each symbol of the codeword is represented by an l-dimensional vector over F and is placed on a separate storage node. We assume that the code is MDS, i.e., the entire codeword can be recovered from any k of its coordinates (from

the encoding stored on any k out of the n nodes). According to the cut-set bound of [5], the amount of information required for repair of a single node from d helper nodes satisfies the inequality

$$\beta(n,k) \geq \frac{dl}{d-k+1}, \tag{1}$$

where $k \leq d \leq n-1$.

Suppose that information is encoded with an MDS array code $\mathcal{C}$ of length $n = \bar{n}u$ over a finite field F. If the size of the code is q^{kl}, we refer to it as a $\mathcal{C}(n,k,l)$ code. The set of nodes $[n] = \{1,2,\ldots,n\}$ is partitioned into $\bar{n}$ subsets (racks) of size u each. Accordingly, the coordinates of the codeword $c \in \mathcal{C}$ are partitioned into segments of length u, and we label them as $c_t, t = 1,\ldots,n$, where $t = (m-1)u + j$, $1 \leq m \leq \bar{n}$, $1 \leq j \leq u$. We do not distinguish between the nodes and the coordinates of the codeword, and refer to both of them as nodes. Each node is an element in F^l, and when needed, we denote its entries as $c_{t,j}$, $j = 1,\ldots,l$.

Denote by $\mathcal{R} \subset \{1,\ldots,\bar{n}\}$ the set of $\bar{d}$ helper racks and let m^* be the index of the host rack. To repair the failed node, information is generated in the helper racks and is combined with the contents of the local nodes to perform the repair. This is modeled by computing a linear function of the contents of the nodes within each helper rack (the function depends on the contents of all the nodes in the rack, and can in principle also depend on the rack index), and sending this information to rack m^*.

Definition II.1 (REPAIR SCHEME)**.** *Let $\mathcal{C}(n,k,l)$ be an array code. Suppose that node $c_{(m^*-1)u+j^*}$ is erased (has failed). To recover the lost data, we rely on the values of the symbols in coordinates c_{iu+j}, where $i \in \mathcal{R}$ and $j = 1,\ldots,u$. A* repair scheme $\mathcal{S}$ *with repair degree $\bar{d} \leq \bar{n}-1$ is formed of $\bar{d}$ functions $f_i : F^{ul} \to F^{\beta_i}, i \in \mathcal{R}$ and a function $g : F^{\sum_{i\in\mathcal{R}}\beta_i} \times F^{(u-1)l} \to F^l$. For a given $i \in \mathcal{R}$ the function f_i maps $c^{(i)}$ (the nodes in rack i) to some β_i symbols of F. The function g accepts these symbols together with the available nodes in the host rack as arguments, and returns the value of the failed node*:

$$g(\{f_i(c_{(i-1)u+j}, 1 \leq j \leq u), i \in \mathcal{R}\}, \{c_{(m^*-1)u+j}, j \in \{1,\ldots,u\}\backslash\{j^*\}\}) = c_{(m^*-1)u+j^*}. \tag{2}$$

In general the function $f_i, i \in \mathcal{R}$ depends on i, m^* and j^*, and the function g depends on $\mathcal{R}, m^*, j^*$.

The quantity $\beta(\mathcal{R}, m^*, j^*) = \sum_{i\in\mathcal{R}}\beta_i$ is called the repair bandwidth of the node $c_{(m^*-1)u+j^*}$ from the helper racks in $\mathcal{R}$ and from the available nodes in the host rack m^*.

The repair scheme can be defined in a more general way: for instance, each of the functions f_i that form the information downloaded by the failed node could depend on the entire set $\mathcal{R}$ (and not just on the contents of the node i) and the function g could depend on the labels of the helper nodes in addition to the information downloaded from them. At the same time, all our results as well as all the results in the earlier literature are well described by this definition, which therefore suffices for our purposes. If the functions f_i, g are F-linear, the repair scheme itself is called *linear*. Only such schemes will be considered below.

Let

$$\beta(n,k,u) := \min_{\mathcal{C}\subset F^{nl}} \max_{\mathcal{R},m^*,j^*} \beta(\mathcal{R}, m^*, j^*)$$

where the minimum is taken over all $(n, M = q^{kl})$ MDS array codes and the maximum over the index of the host rack, the failed node in the rack, and the choice of the set of the helper racks $\mathcal{R}$. To rule out the trivial case, we assume throughout that $k \geq u$.

A. Optimal Repair

Suppose that $k = \bar{k}u + v$, where $0 \leq v \leq u-1$. A necessary condition for successful repair of a single node is given by a version of the cut-set bound [9], [8] which states that for any (n,k,l) MDS array code, the (inter-rack) repair bandwidth is at least

$$\beta(n,k,u) \geq \frac{\bar{d}l}{\bar{d}-\bar{k}+1} \tag{3}$$

The code that attains this bound with equality is said to have the optimal repair property.

The arguments below are based on the following obvious (and well-known) observation.

Lemma II.2. *Let $\mathcal{C}(n,k,l)$ be an MDS array code. Suppose that a failed node is repaired using a set $\mathcal{I}$, $|\mathcal{I}| = d$ of helper nodes. The number of symbols of F downloaded for the repair task from any subset $\mathcal{I}' \subset \mathcal{I}$ of size $|\mathcal{I}'| = d-k+1$ is at least l.*

To prove this it suffices to observe that, because of the MDS property, no subset of $k-1$ nodes carries any information about the value of any other node.

We note that this lemma applies to the rack model (i.e., allowing processing of the information obtained from the nodes in $\mathcal{I}$). It also applies if the count of downloaded symbols is replaced by the count of symbols *accessed* on the helper nodes.

The next statement, called the *uniform download property*, is well known for the case of homogeneous storage. Its proof for the rack-aware storage is not much different, and is given for completeness in Appendix A.

Proposition II.3. *Let $\mathcal{C}$ be an MSR code and suppose that $\bar{k} > 1$. Let $\mathcal{R}$ be the set of helper racks used to repair a single failed node. Then $\beta_i = l/(\bar{d}-\bar{k}+1), i \in \mathcal{R}$.*

We note that both the bound (3) and this proposition can be generalized to the case of $2 \leq h \leq r$ failed nodes located on the same rack without any difficulty; for instance, the bound takes the form $\beta \geq \frac{h\bar{d}l}{\bar{d}-\bar{k}+1}$.

Next, observe that if $\bar{k}$ is divisible by the rack size u, then any MSR code for the standard model will be optimal for the rack model, i.e., cooperation between the nodes within the rack does not help to reduce the repair bandwidth (this has been first observed in [8, Thm. 4]).

Proposition II.4. *Let $k = \bar{k}u$, and let $\mathcal{C}$ be an MSR code of length $n = \bar{n}u$ with optimal repair of a single node for the homogeneous storage model. Then $\mathcal{C}$ attains the cut-set bound* (3) *for repair of any single node in the rack-aware model.*

Proof: Take an MSR code of length n and assume that $\upsilon = 0$. Suppose that the number of helper nodes is d, and this includes the $u-1$ local nodes. By (1), the repair bandwidth necessary equals $\frac{d}{d-k+1}l$. In accordance with the model, take $d = \bar{d}u + (u-1)$, then

$$\frac{d}{d-k+1}l = \Big(\frac{\bar{d}}{\bar{d}-\bar{k}+1} + \frac{u-1}{d-k+1}\Big)l \tag{4}$$

and this achieves the bound (3) if the second term is discounted (which is possible because of the uniform download property and because intra-rack communication is free). □

Note that in the case of $\upsilon \neq 0$, optimal codes for the rack model perform repair using a strictly smaller repair bandwidth than optimal codes for the homogeneous model. This also suggests that the number of symbols downloaded from a helper rack is strictly smaller than the number of accessed symbols, i.e., intra-rack processing is necessary for optimal repair (this will be made rigorous once we establish Prop. II.6 below).

For reader's convenience, let us summarize the code parameters: We consider (n,k,l) array codes used in a system where the nodes are arranged in racks of size u. The codes are designed to repair a single node. We further assume that $n = \bar{n}u, k = \bar{k}u + \upsilon$, where $0 < \upsilon \leq u-1$, and the number of helper racks is $\bar{d}$, where $\bar{k} \leq \bar{d} \leq \bar{n}-1$. We also use the notation $r = n-k, \bar{r} = \bar{n}-\bar{k}$ for the number of parity nodes and parity racks, respectively. Finally, to shorten the formulas we denote

$$s = d-k+1, \quad \bar{s} = \bar{d}-\bar{k}+1,$$

where d is the total number of helper nodes accessed for repair, and $\bar{d}$ is the *repair degree*, i.e., number of helper racks (not counting the host rack).

B. Optimal Access

Some of the constructions of codes for the homogeneous case have the additional property that the information accessed on the helper nodes is the same as the information that is downloaded by the helper node (no processing is performed before downloading). This property, also called *repair by transfer*, reduces the implementation overhead, and is therefore desirable in the code construction. Structure and constructions of optimal access (OA) codes for the homogeneous case were addressed in [24], [27], [30] among others.

Definition II.5. *Let $\mathcal{C}(n = \bar{n}u, k, l)$ be a code that supports optimal repair of a single failed node with repair degree $\bar{d}$. Suppose that each of the helper racks provides $l/\bar{s}$ field symbols and these symbols are generated by accessing the smallest possible number of symbols of the nodes in the rack. In this case we say that $\mathcal{C}$ has the OA property.*

To motivate this definition, we draw an analogy with the homogeneous case. In this case, on account of the bound (1) and the uniform download property, the system accesses l/s symbols at each of the helper nodes, and these symbols are downloaded to accomplish the repair. As a consequence, a group of $u > 1$ helper nodes provides ul/s symbols. This observation also extends to the rack-aware model in the case that $u|k$. Indeed, in this case the number of symbols downloaded from, and accessed on, each rack equals $l/\bar{s} = ul/s$.

In the next proposition (proved in Appendix B) we derive a lower bound on the number of accessed symbols and establish the *uniform access condition*.

Proposition II.6. *Let $\mathcal{C}$ be an (n,k,l) optimal-repair MDS array code for the rack model with repair degree $\bar{d} \geq \bar{k}+1$ and $u \leq k$. The total number of symbols accessed on the helper racks for repair of a single node satisfies*

$$\alpha \geq \frac{\bar{d}ul}{s}. \tag{5}$$

Equality holds if and only if the number of symbols accessed on node e satisfies $\alpha_{m,e} = l/s$ for all $m \in \mathcal{R}$; $e = 1,\ldots,u$.

As noted above, if $u|k$, the symbols accessed on the helper nodes can be downloaded without processing, accounting for optimal repair. At the same time, if $u \nmid k$, and the code meets the bound (5), then processing is necessary because $\bar{d}ul/s$ is strictly greater than the optimal bandwidth in (3).

C. A Lower Bound on the Sub-Packetization of Rack-Aware Optimal-Access MSR Codes

In this section we present a lower bound on the value of the node size in MSR codes for the rack model, which will be implicitly assumed throughout without further mention. Similarly to [3], [24], we limit ourselves to systematic codes and linear repair schemes. Let $\mathcal{C}$ be an $(n = \bar{n}u, k = \bar{k}u, l)$ systematic optimal-access MSR array code over F. Let $A = (A_{ij})$ be the $((n-k)l \times kl)$ encoding matrix of $\mathcal{C}$; in other words, the parity symbols $c_{k+i}, i = 1\ldots, r = n-k$ are obtained from the data symbols c_j, $j = 1,\ldots,k$ according to the relation

$$c_{k+i} = \sum_{j=1}^{k} A_{i,j}c_j, \tag{6}$$

where each $A_{i,j}$ is an $l \times l$ invertible matrix over F. Assume without loss of generality that the k systematic nodes are located on racks $1,\ldots,\bar{k}$, called systematic racks below. Racks $\bar{k}+1,\ldots,\bar{n}$ will be called parity racks. Let $\boldsymbol{c}_m = (c_{(m-1)u+1},\ldots,c_{mu})^T$ be the data vector stored in the m-th rack, $1 \leq m \leq \bar{k}$, where each component is an l-vector over F. Suppose for definiteness that the failed node is located in rack m_1, where $1 \leq m_1 \leq \bar{k}$. Suppose further that the set of $\bar{d}$ helper racks is formed of the remaining $\bar{k}-1$ systematic racks and some $\bar{s} = \bar{d}-\bar{k}+1$ parity racks.

We assume throughout that the repair scheme is independent of the index of the failed node in its rack.

The main result of this section is given in the following theorem, whose proof is modeled on the result of [3] and generalizes its main ideas to the case of $u \geq 2$.

Theorem II.7. *Let $\mathcal{C}$ be an $(n = \bar{n}u, k = \bar{k}u, l)$ optimal-access MSR array code, $k \geq u$, and let $\bar{d}, \bar{k} \leq \bar{d} \leq \bar{n}-1$ be the size of the helper set $\mathcal{R}$. Suppose further that there is a linear repair scheme that supports repair of a single failed node from any $\bar{d}$ helper racks.*

(a) Suppose that the repair scheme depends on the choice of the helper racks as well as on the index of the host rack. Then

$$l \geq \min\{\bar{s}^{(\bar{n}-1)/s}, \bar{s}^{\bar{k}-1}\}, \tag{7}$$

where $\bar{s} = \bar{d} - \bar{k} + 1$ *and* $s = \bar{s}u$.

(b) Suppose that the repair scheme depends on the index of the host rack but not on the choice of the helper racks, then

$$l \geq \min\{\bar{s}^{\bar{n}/s}, \bar{s}^{\bar{k}-1}\}. \tag{8}$$

A proof of this theorem is given in the Appendix. Here let us make the following remark. The theorem is proved under the assumption that $u|k$, in which case any optimal-access MSR code for the homogeneous storage model supports optimal repair for the rack model. The smallest possible value of sub-packetization for such codes is $l = r^{\lceil \frac{n-1}{r} \rceil}$ [3], [30]. Thus, this theorem says that it is possible that there exist optimal-access rack codes that have smaller node size than OA codes for homogeneous storage *even in the case when k is a multiple of u*.

III. Rack-Aware Codes With Optimal Repair for All Parameters

Let $\bar{s} = \bar{d} - \bar{k} + 1$ and let $F, |F| > \bar{s}n$ be a finite field. The code that we construct is formed as an F-linear array MDS code $\mathcal{C}$ of length n, dimension k, and sub-packetization $l = \bar{s}^{\bar{n}}$. We denote a codeword of $\mathcal{C}$ by $(c_1, c_2, \ldots, c_n)$, where $c_i = (c_{i,1}, \ldots, c_{i,l})$ for all $i = 1, \ldots, n$. Suppose that $\bar{s}n \,|\, (|F| - 1)$ and let $\lambda \in F$ be an element of multiplicative order $\bar{s}n$. Finally, given $j \in \{0, 1, \ldots l-1\}$, consider the base $\bar{s}$ expansion $j = (j_{\bar{n}}, j_{\bar{n}-1}, \ldots, j_1)$ and let

$$j(p, a) := (j_{\bar{n}}, \ldots, j_{p+1}, a, j_{p-1}, \ldots, j_1), \tag{9}$$

where $0 \leq a \leq \bar{s} - 1$.

Construction III.1. *Consider an* $(n, k, l = \bar{s}^{\bar{n}})$ *code* $\mathcal{C} = \{\boldsymbol{c} = (c_{i,j})_{1 \leq i \leq n; 0 \leq j \leq l-1}\}$ *defined by the following set of rl parity-check equations over F:*

$$\sum_{e=1}^{\bar{n}} \lambda^{t((e-1)\bar{s}+j_e)} \sum_{i=1}^{u} \lambda^{t(i-1)\bar{s}\bar{n}} c_{(e-1)u+i,j} = 0 \tag{10}$$

for all $t = 0, \ldots, r-1$; $j = 0, \ldots, l-1$.

We will show that the code defined in (10) is an MDS code that has the smallest possible repair bandwidth according to the bound (3). Before stating the main theorem that proves these claims let us comment on the origin as well as the new elements in this construction. The code is formed of two levels, the algebraic one, which accounts for the repair of a node in any *fixed* rack, say $p, 1 \leq p \leq \bar{n}$, and a stacking construction which makes the code universal (i.e., rack-independent). The second part is accomplished by representing the index j of the parity check equation as an $\bar{s}$-ary number (9). This expansion enables us to isolate the parities that are used to perform repair of any failed node in rack p, specifically, they are the equations in (10) whose label j is obtained by varying the value of the entry j_p in the expansion (9) and fixing all the remaining values.

The algebraic development represents the main part of the proof of Theorem III.1 and accounts for the optimal-bandwidth repair scheme. The key new idea utilized in the proof is the choice of λ based on the multiplicative structure of F and using the evaluation points given by the powers of λ.

Theorem III.1. *Let* $\bar{k} \leq \bar{d} \leq \bar{n} - 1$. *The* $(n, k, l = \bar{s}^{\bar{n}})$ *code* $\mathcal{C}$ *defined by the parity-check equations* (10) *is an MDS code that supports optimal repair of any single node from any* $\bar{d}$ *helper racks, under the rack-aware storage model.*

Proof: We begin with proving the part of the claim about the repair properties of the code $\mathcal{C}$. Suppose that the index of the rack that contains the failed node is $p \in \{1, \ldots, \bar{n}\}$. We have $\bar{r}u = r + v$ and since $0 \leq v \leq u - 1$, $(\bar{r} - 1)u \leq r - 1$. Rewriting (10), we have:

$$\lambda^{t((p-1)\bar{s}+j_p)} \sum_{i=1}^{u} \lambda^{t(i-1)\bar{s}\bar{n}} c_{(p-1)u+i,j} = -\sum_{\substack{e=1 \\ e \neq p}}^{\bar{n}} \lambda^{t((e-1)\bar{s}+j_e)} \sum_{i=1}^{u} \lambda^{t(i-1)\bar{s}\bar{n}} c_{(e-1)u+i,j}$$

for all $t = 0, \ldots, r-1$; $j = 0, \ldots, l-1$. We will use a subset of the parity-check equations with indices t of the form $t = wu$:

$$\lambda^{((p-1)\bar{s}+j_p)wu} \sum_{i=1}^{n} c_{(p-1)u+i,j} = -\sum_{e \neq p} \lambda^{((e-1)\bar{s}+j_e)wu} \sum_{i=1}^{u} c_{(e-1)u+i,j}$$

for all $j = 0, \ldots, l-1$; $w = 0, 1, \ldots, \bar{r} - 1$, where we have used the fact that $\lambda^{\bar{s}n} = 1$. Denoting $\alpha = \lambda^u$ and summing these equations on $j_p = 0, 1, \ldots, \bar{s} - 1$, we obtain the following set of conditions:

$$\sum_{j_p=0}^{\bar{s}-1} \alpha^{((p-1)\bar{s}+j_p)w} \sum_{i=1}^{u} c_{(p-1)u+i,j} = -\sum_{e \neq p} \alpha^{((e-1)\bar{s}+j_e)w} \sum_{j_p=0}^{\bar{s}-1} \sum_{i=1}^{u} c_{(e-1)u+i,j} \tag{11}$$

for all $w = 0, 1, \ldots, \bar{r} - 1$ and all $j_{\bar{n}}, \ldots, j_{p+1}, j_{p-1}, \ldots, j_1$, where each of these values ranges over $\{0, 1, \ldots, \bar{s} - 1\}$. Let $\mathcal{R} = \{q_1, \ldots, q_{\bar{d}}\}$ be the set of helper racks and let $[\bar{n}] \setminus \mathcal{R} = \{p, p_1, \ldots, p_{\bar{r}-\bar{s}}\}$. Then (11) can be written as follows:

$$\sum_{j_p=0}^{\bar{s}-1} \alpha^{((p-1)\bar{s}+j_p)w} \sum_{i=1}^{u} c_{(p-1)u+i,j} + \sum_{\substack{a \in [\bar{n}] \setminus \mathcal{R} \\ a \neq p}} \alpha^{((a-1)\bar{s}+j_a)w} \sum_{j_p=0}^{\bar{s}-1} \sum_{i=1}^{u} c_{(a-1)u+i,j} = -\sum_{b \in \mathcal{R}} \alpha^{((b-1)\bar{s}+j_b)w} \sum_{j_p=0}^{\bar{s}-1} \sum_{i=1}^{u} c_{(b-1)u+i,j}. \tag{12}$$

$$\begin{bmatrix} 1 & \dots & 1 & 1 & \dots & 1 \\ \alpha^{\bar{s}(p-1)} & \dots & \alpha^{\bar{s}(p-1)+\bar{s}-1} & \alpha^{\bar{s}(p_1-1)+j_{p_1}} & \dots & \alpha^{\bar{s}(p_{\bar{r}-\bar{s}}-1)+j_{p_{\bar{r}-\bar{s}}}} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ (\alpha^{\bar{s}(p-1)})^{\bar{r}-1} & \dots & (\alpha^{\bar{s}(p-1)+\bar{s}-1})^{\bar{r}-1} & (\alpha^{\bar{s}(p_1-1)+j_{p_1}})^{\bar{r}-1} & \dots & (\alpha^{\bar{s}(p_{\bar{r}-\bar{s}}-1)+j_{p_{\bar{r}-\bar{s}}}})^{\bar{r}-1} \end{bmatrix} \begin{bmatrix} \sum_{i=1}^{u} c_{(p-1)u+i,j(p,0)} \\ \vdots \\ \sum_{i=1}^{u} c_{(p-1)u+i,j(p,\bar{s}-1)} \\ \sigma_{p_1,j(p,*)} \\ \vdots \\ \sigma_{p_{\bar{r}-\bar{s}},j(p,*)} \end{bmatrix}$$

$$= -\begin{bmatrix} 1 & \dots & 1 \\ \alpha^{\bar{s}(q_1-1)+j_{q_1}} & \dots & \alpha^{\bar{s}(q_{\bar{d}}-1)+j_{q_{\bar{d}}}} \\ \vdots & \vdots & \vdots \\ \alpha^{(\bar{s}(q_1-1)+j_{q_1})(\bar{r}-1)} & \dots & \alpha^{(\bar{s}(q_{\bar{d}}-1)+j_{q_{\bar{d}}})(\bar{r}-1)} \end{bmatrix} \begin{bmatrix} \sigma_{q_1,j(p,*)} \\ \vdots \\ \sigma_{q_{\bar{d}},j(p,*)} \end{bmatrix} \tag{13}$$

$$|\lambda^{j_1}, \lambda^{j_1+\bar{s}\bar{n}}, \dots, \lambda^{j_1+(u-1)\bar{s}\bar{n}} | \lambda^{j_2+\bar{s}}, \lambda^{j_2+\bar{s}(1+\bar{n})}, \dots, \lambda^{j_2+\bar{s}(1+(u-1)\bar{n})} | \dots | \lambda^{j_{\bar{n}}+\bar{s}(\bar{n}-1)}, \lambda^{j_{\bar{n}}+\bar{s}(2\bar{n}-1)}, \dots, \lambda^{j_{\bar{n}}+\bar{s}(\bar{n}-1+(u-1)\bar{n})}| \tag{14}$$

In matrix form these equations are shown in (13) above, where

$$\sigma_{e,j(p,*)} := \sum_{j_p=0}^{\bar{s}-1} \sum_{i=1}^{u} c_{(e-1)u+i,j}, \quad e = 1, \dots, \bar{n},$$

and j is as given above after (11).

We claim that Equations (13) suffice to recover one failed node in rack p. Indeed, suppose that the $\bar{d}$-dimensional vector on the right-hand side of (13) is made available to the failed node by transmitting one symbol of F from each of the helper racks. Let us check that the matrix on the left-hand side is Vandermonde, i.e., that the defining elements in the second row are distinct. To see this, note that $\text{ord}(\alpha) = \bar{s}\bar{n}$, and the maximum degree of α in the set $\{\alpha^{\bar{s}(e-1)+m}, m = 0, \dots, \bar{s}-1; a = 1, \dots, \bar{n}\}$ is

$$\bar{s}(\bar{n}-1) + \bar{s} - 1 < \bar{s}\bar{n}.$$

Moreover, each of the first $\bar{s}$ coordinates of the multiplier vector on the left-hand side of (13)

$$\Big(\sum_{i=1}^{u} c_{(p-1)u+i,j(p,0)}, \dots, \sum_{i=1}^{u} c_{(p-1)u+i,j(p,\bar{s}-1)} \Big)^T$$

contains only one unknown term which corresponds to the failed node. Thus, if the values $c_{(p-1)u+i,j(p,*)}$ of all the functional local nodes are made available to the failed node (recall that this does not count toward the repair bandwidth), then system (13) can be solved to find the entries of the missing node. This calculation is repeated $\bar{s}^{\bar{n}-1}$ times for each assignment of the values $j_{\bar{n}}, \dots, j_{p+1}, j_{p-1}, \dots, j_1$, thereby completing the repair procedure.

Let us compute the inter-rack repair bandwidth of the described procedure. To repair the entries of the single failed node in the pth rack with indices in the subset $\{j(p,a), a = 0, 1, \dots, \bar{s}-1\}$ we download one symbol of F from each of the $\bar{d}$ helper racks. There are $\bar{s}^{\bar{n}-1}$ subsets of the above form, and thus the total repair bandwidth is

$$\bar{d}\bar{s}^{\bar{n}-1} = \frac{\bar{d}l}{\bar{s}},$$

proving the optimality claim of the code according to (3).

Finally let us prove that the code $\mathcal{C}$ is MDS. This is immediate upon observing that each subset of parity-check equations isolated by fixing the value of $j = 0, 1, \dots, l-1$ defines an MDS code. To check this, observe that the set of rows of the parity-check matrix of $\mathcal{C}$ for a fixed value $j = (j_{\bar{n}}, \dots, j_1)$ forms a set of parities of a generalized Reed-Solomon codes (i.e., each column is a set of powers of an element of F), and the defining row of this set of parities is shown in (14), as shown at the top of this page, where each group between the vertical bars corresponds to a fixed value of $s = 1, \dots, \bar{n}$ in (10). It suffices to show that all these elements are distinct or that these groups do not overlap. Note that the largest power in (14) is

$$j_{\bar{n}} + \bar{s}(\bar{n}-1+(u-1)\bar{n}) \le \bar{s}-1+u\bar{n}\bar{s}-\bar{s} < \bar{s}n = \text{ord}(\lambda). \tag{15}$$

Now consider two groups and let their numbers be a and b, where $1 \le b < a \le \bar{n}$. Then the difference between the exponents of the first elements in the two groups is

$$(a-b)\bar{s} + (j_a - j_b) \ge 1$$

so the first elements are obviously distinct. Further, the exponents of the elements in each of the groups are obtained by adding a multiple of $\bar{s}\bar{n}$ to the exponent of the first element, which together with (15) implies that the groups are disjoint. This shows that the code $\mathcal{C}$ is MDS, and the proof is complete. ■

We remark that the repair procedure relies on a subset of the parity-check equations of the code $\mathcal{C}$. Namely, the only rows of the parity-check matrix that we use are the rows whose numbers are integer multiples of the size of the rack u. It suffices to use only these parities because the assumptions of the rack model are relaxed compared to the standard definition of regenerating codes. The remaining parities support the MDS property of the code $\mathcal{C}$ and do not contribute to the repair procedure.

In Sec. IV-B we construct codes with somewhat better parameters than the codes given by Construction III.1. Specifically, the smallest field size required for the code family in Sec. IV-B is $n + \bar{s} - 1$ (as opposed to $\bar{s}n$), and the repair procedure accesses fewer symbols on the helper nodes than the procedure presented in the above proof. At the same time, the codes presented in this section have the *optimal update property.* Namely, a codeword of the code $\mathcal{C}$ can be viewed as an $l \times n$ array, and for a given row index $j \in \{1, \dots, l-1\}$

the n symbols are encoded with a generalized RS code independently of the other rows. Thus, if some k symbols are taken as information symbols, then the change of one symbol in the data requires to change r parity symbols, which is also the smallest possible number [24]. At the same time, the codes in the family of Sec. IV-B do not have optimal update, and are in this respect inferior to the present construction.

IV. Low-Access Codes for the Rack Model

This section aims at constructing an optimal-repair MSR code for the rack model that accesses a reduced number of symbols on the nodes in the helper racks. Our presentation is formed of two parts. In the first part we construct an optimal-access MSR code for arbitrary repair degree $k \le d \le n-1$ *without assuming the rack model* of storage. The code has subpacketization $l = (d-k+1)^n$.

In the second part we present a modification of this construction for the rack model, attaining subpacketization $l = \bar{s}^{\bar{n}}$. Note that this value is smaller than the smallest node size of known constructions of OA codes for the homogeneous model, which is s^n [30].

A. Optimal-Access MSR Codes With Arbitrary Repair Degree for Homogeneous Storage

In this section we present a family of OA codes for any repair degree $k \le d \le n-1$. Let $s = d-k+1$ and let $F, |F| \ge n+s-1$ be a finite field. Let $\lambda_0, \ldots, \lambda_{n-1}, \mu_1, \ldots, \mu_{s-1}$ be $n+s-1$ distinct elements of F. Let $i = (i_{n-1}, \ldots, i_0)$ be the s-ary representation of $i = 0, \ldots, l-1$ and (as before) let $i(a,b) = (i_{n-1}, \ldots, i_{a+1}, b, i_{a-1}, \ldots, i_0)$ for $0 \le a \le n-1$ and $0 \le b \le s-1$. For brevity below we use the notation

$$\delta(i) := \mathbb{1}_{\{i=0\}}.$$

Construction IV.1. *Define an $(n, k = n-r, l = s^n)$ array code $\mathcal{C} = \{\boldsymbol{c} = (c_{j,i})_{0\le j\le n-1; 0\le i\le l-1}\}$, where the codeword $\boldsymbol{c}$ satisfies the following parity check equations over F:*

$$\sum_{j=0}^{n-1} \lambda_j^t c_{j,i} + \sum_{j=0}^{n-1} \delta(i_j) \sum_{p=1}^{s-1} \mu_p^t c_{j,i(j,p)} = 0, \quad i = 0, \ldots, l-1;\ t = 0, \ldots, r-1. \tag{16}$$

Since later in this section we rely on multiplicative structure of F, we label the nodes $0, \ldots, n-1$ and not $1, \ldots, n$ as in Construction III.1. In the next subsection we will also label the racks from 0 to $\bar{n}-1$ for the same reason.

Theorem IV.1. *The code $\mathcal{C}$ defined in* (16) *is an optimal-access MDS array code.*

The proof will be omitted because in principle it can be obtained from the proof of Theorem IV.2 below upon taking the size of the rack $u = 1$. This is however not entirely immediate, and interested readers can consult the arXiv posting of a preprint of this paper (arXiv:1901.04419, January 2019) which contains a complete and independent proof of Theorem IV.1.

B. Rack-Aware MSR Codes With Low Access

In this section we adapt the code family constructed in Sec. IV-A for the rack-aware storage model. This result is obtained by adjusting the sub-packetization and by carefully choosing the elements $\lambda_0, \ldots, \lambda_{n-1}$.

We aim to construct an (n, k, l) MDS array code over F, where $n = \bar{n}u$, and u is the size of the rack. Recall that $\bar{s} = \bar{d} - \bar{k} + 1$ where $\bar{k} \le \bar{d} \le \bar{n} - 1$, and $\bar{k} = \lfloor k/u \rfloor$. Let $|F| \ge n + \bar{s} - 1$ and $n | (|F| - 1)$. Let $\lambda \in F$ be an element of multiplicative order n, and let $\mu_1, \ldots, \mu_{\bar{s}-1}$ be $\bar{s}-1$ distinct elements in $F \setminus \{\lambda^i \mid i = 0, \ldots, n-1\}$. For $j = 0, \ldots, n-1$, let us write $j = eu + g$ where $0 \le e < \bar{n}$ and $0 \le g < u$.

We construct an rack-aware low-access MSR code over F that can repair any single node from any $\bar{d}$ helper racks.

Construction IV.2. *Define an $(n, k = n-r, l = \bar{s}^{\bar{n}})$ array code $\mathcal{C} = \{(c_{j,i})_{0\le j\le n-1; 0\le i\le l-1}\}$ by the following parity-check equations over F:*

$$\sum_{j=0}^{n-1} \lambda_j^t c_{j,i} + \sum_{j=0}^{n-1} \delta(i_e) \sum_{p=1}^{\bar{s}-1} \mu_p^t c_{j,i(e,p)} = 0, \tag{17}$$

where $\lambda_j = \lambda^{e+g\bar{n}}$, $i = 0, \ldots, l-1$ and $t = 0, \ldots, r-1$.

We will show that this code family supports optimal repair while accessing $l/\bar{s}$ symbols on each of the nodes in the helper racks, which is by a factor of $s/\bar{s} \approx u$ greater than the bound in Prop. II.6. While these codes stop short of attaining the bound (5), they have lower access requirement than the codes given by Construction III.1, which access all symbols of the helper nodes, i.e., $\bar{s}$ times more symbols than the current construction.

Theorem IV.2. *The code $\mathcal{C}$ defined in* (17) *is an optimal-repair MDS array code. The repair procedure accesses $l/\bar{s}$ symbols on each of the nodes in $\bar{d}$ helper racks. The repair scheme does not depend on the choice of the subset of $\bar{d}$ helper racks.*

Proof: I. Optimal-access property. Suppose c_{j_1} is the failed node, where $j_1 = e_1\ u + g_1$. Let $\mathcal{R}$ be the set of helper racks and let $\mathcal{J} = \{0, \ldots, \bar{n}-1\} \setminus \mathcal{R}$. We write this set as $\mathcal{J} = \{e_1, e_2, \ldots, e_{\bar{n}-\bar{d}}\}$. For a given a, $1 \le a \le \bar{n} - \bar{d}$ we will need a-subsets of $\mathcal{J}$, which we denote by $\mathcal{J}_a$. We always assume that $e_1 \in \mathcal{J}_a$. As before, let $\mathcal{I} \subset \{0, 1, \ldots, l-1\}$ be the subset of indices such that $i_{e_1} = 0$; let

$$\mathcal{I}_1 = \{i = (i_{\bar{n}-1}, \ldots, i_0) \in \{0, \ldots, l-1\} \mid i_{e_1} = 0; i_e \ne 0, e \in \mathcal{J} \setminus \mathcal{J}_1\}$$

and define

$$\mathcal{I}_a = \bigcup_{\mathcal{J}_a \subseteq \mathcal{J}} \mathcal{I}(\mathcal{J}_a), \quad a = 2, \ldots, \bar{n} - \bar{d},$$

where

$$\mathcal{I}(\mathcal{J}_a) = \{i = (i_{\bar{n}-1}, \ldots, i_0) \in \{0, \ldots, l-1\} \mid i_e = 0, e \in \mathcal{J}_a; i_e \ne 0, e \in \mathcal{J} \setminus \mathcal{J}_a\}.$$

Recall that $\bar{r} = \bar{n} - \bar{k}$. We will use the parity check equations corresponding to $i \in \mathcal{I}$ and all powers $t = uw$, $w = 0, \ldots, \bar{r}-1$ to repair c_{j_1}. To show that the repair is possible, we argue by induction on $a = 1, \ldots, \bar{n} - \bar{d}$.

To prove the induction basis, we show that it is possible to recover the values $\{c_{j_1,i(e_1,p)} \mid p = 0,\dots,\bar{s}-1\}$ and $\{\sum_{g=0}^{u-1} c_{eu+g,i} \mid e \in \mathcal{J} \setminus \mathcal{J}_1\}$ for every $i \in \mathcal{I}_1$ from the helper racks $\mathcal{R}$. From (17), for $i \in \mathcal{I}_1$, we have

$$\sum_{e\in\mathcal{J}}\sum_{g=0}^{u-1}\lambda_{eu+g}^{t}c_{eu+g,i}+\sum_{g=0}^{u-1}\sum_{p=1}^{\bar{s}-1}\mu_p^t c_{e_1\ u+g,i(e_1,p)}$$
$$=-\sum_{e\in\mathcal{R}}\sum_{g=0}^{u-1}\Big(\lambda_{eu+g}^{t}c_{eu+g,i}+\delta(i_e)\sum_{p=1}^{\bar{s}-1}\mu_p^t c_{eu+g,i(e,p)}\Big).$$

Using $t = uw$, $\lambda_{eu+g} = \lambda^{e+g\bar{n}}$, and $\lambda^{\bar{n}u} = 1$, we obtain

$$\sum_{e\in\mathcal{J}}\lambda^{euw}\sum_{g=0}^{u-1}c_{eu+g,i}+\sum_{p=1}^{\bar{s}-1}\mu_p^{uw}\sum_{g=0}^{u-1}c_{e_1u+g,i(e_1,p)}$$
$$=-\sum_{e\in\mathcal{R}}\Big(\lambda^{euw}\sum_{g=0}^{u-1}c_{eu+g,i}+\delta(i_e)\sum_{p=1}^{\bar{s}-1}\mu_p^{uw}\sum_{g=0}^{u-1}c_{eu+g,i(e,p)}\Big), \tag{18}$$

$i \in \mathcal{I}_1$, $w = 0,\dots,\bar{r}-1$. To shorten our notation, denote the right-hand side of (18) by $\sigma_{i,w}(\mathcal{J}_1)$ and let

$$\pi_{i,e} := \sum_{g=0}^{u-1} c_{eu+g,i}.$$

Note that the value of $\sigma_{i,w}(\mathcal{J}_1)$ only depends on the helper racks. For $i = 1,\dots,\bar{n}-\bar{d}$ define $\alpha_i := \lambda^{e_i u}$. Let us write equations (18) for all $w = 0,\dots,\bar{r}-1$ in matrix form:

$$\begin{bmatrix} 1 & 1 & \cdots & 1 & 1 & \cdots & 1 \\ \alpha_1 & \mu_1 & \cdots & \mu_{\bar{s}-1} & \alpha_2 & \cdots & \alpha_{\bar{n}-\bar{d}} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ \alpha_1^{\bar{r}-1} & \mu_1^{\bar{r}-1} & \cdots & \mu_{\bar{s}-1}^{\bar{r}-1} & \alpha_2^{\bar{r}-1} & \cdots & \alpha_{\bar{n}-\bar{d}}^{\bar{r}-1} \end{bmatrix}$$
$$\times\begin{bmatrix} \pi_{i,e_1} \\ \pi_{i(e_1,1),e_1} \\ \vdots \\ \pi_{i(e_1,\bar{s}-1),e_1} \\ \pi_{i,e_2} \\ \vdots \\ \pi_{i,e_{\bar{n}-\bar{d}}} \end{bmatrix} = \begin{bmatrix} \sigma_{i,0}(\mathcal{J}_1) \\ \sigma_{i,1}(\mathcal{J}_1) \\ \vdots \\ \sigma_{i,\bar{r}-1}(\mathcal{J}_1) \end{bmatrix}. \tag{19}$$

Observe that the matrix on the left-hand side of (19) is invertible. Therefore, the values $\{c_{j_1,i(j_1,p)} \mid p = 0,\dots,\bar{s}-1\}$ and $\{\sum_{g=0}^{u-1} c_{eu+g,i} \mid e \in \mathcal{J}\setminus\mathcal{J}_1\}$ can be found from the values $\{\sigma_{i,w}(\mathcal{J}_1) \mid w = 0,\dots,\bar{r}-1\}$ and the local nodes $\{c_{e_1\ u+g} \mid g \neq g_1\}$ for every $i \in \mathcal{I}_1$. This completes the proof of the induction basis.

Now let us fix $a \in \{2,\dots,\bar{n}-\bar{d}\}$ and suppose that we have recovered the values $\{c_{j_1,i(e_1,p)} \mid p = 0,\dots,\bar{s}-1\}$ and $\left\{\sum_{g=0}^{u-1} c_{eu+g,i} \mid e \in \mathcal{J}\setminus\mathcal{J}_1\right\}$, $i \in \mathcal{I}_{a'}$; $1 \le a' \le a-1$ from the information downloaded from the helper racks $\mathcal{R}$.

Fix a subset $\mathcal{J}_a$, $|\mathcal{J}_a| = a$, and let $i \in \mathcal{I}(\mathcal{J}_a)$. From (17), we have

$$\sum_{e\in\mathcal{J}}\sum_{g=0}^{u-1}\lambda_{eu+g}^{t}c_{eu+g,i}+\sum_{e\in\mathcal{J}_a}\sum_{g=0}^{u-1}\sum_{p=1}^{\bar{s}-1}\mu_p^t c_{eu+g,i(e,p)}$$
$$=\sum_{e\in\mathcal{J}}\sum_{g=0}^{u-1}\lambda_{eu+g}^{t}c_{eu+g,i}+\sum_{p=1}^{\bar{s}-1}\mu_p^t\sum_{e\in\mathcal{J}_a}\sum_{g=0}^{u-1}c_{eu+g,i(e,p)}$$
$$=-\sum_{e\in\mathcal{R}}\Big(\sum_{g=0}^{u-1}\lambda_{eu+g}^{t}c_{eu+g,i}+\delta(i_e)\sum_{p=1}^{\bar{s}-1}\mu_p^t\sum_{g=0}^{u-1}c_{eu+g,i(e,p)}\Big). \tag{20}$$

Using $t = uw$, $\lambda_{eu+g} = \lambda^{e+g\bar{n}}$, and $\lambda^{\bar{n}u} = 1$, we obtain

$$\sum_{e\in\mathcal{J}}\lambda^{euw}\sum_{g=0}^{u-1}c_{eu+g,i}+\sum_{p=1}^{\bar{s}-1}\mu_p^{uw}\sum_{e\in\mathcal{J}_a}\sum_{g=0}^{u-1}c_{e_1u+g,i(e_1,p)}$$
$$=-\sum_{e\in\mathcal{R}}\Big(\lambda^{euw}\sum_{g=0}^{u-1}c_{eu+g,i}+\delta(i_e)\sum_{p=1}^{\bar{s}-1}\mu_p^{uw}\sum_{g=0}^{u-1}c_{eu+g,i(e,p)}\Big). \tag{21}$$

Again for notational convenience denote the right-hand side of (21) by $\sigma_{i,w}(\mathcal{J}_a)$ and let

$$\rho_{i,p} := \sum_{e\in\mathcal{J}_a}\sum_{g=0}^{u-1}c_{eu+g,i(e,p)}.$$

Note that the value of $\sigma_{i,w}(\mathcal{J}_a)$ depends only on the information in the helper racks. Let us write Equations (21) for all $w = 0,\dots,\bar{r}-1$ in matrix form:

$$\begin{bmatrix} 1 & \cdots & 1 & 1 & \cdots & 1 \\ \mu_1 & \cdots & \mu_{\bar{s}-1} & \alpha_1 & \cdots & \alpha_{\bar{n}-\bar{d}} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ \mu_1^{\bar{r}-1} & \cdots & \mu_{\bar{s}-1}^{\bar{r}-1} & \alpha_1^{\bar{r}-1} & \cdots & \alpha_{\bar{n}-\bar{d}}^{\bar{r}-1} \end{bmatrix}$$
$$\times\begin{bmatrix} \rho_{i,1} \\ \vdots \\ \rho_{i,\bar{s}-1} \\ \pi_{i,e_1} \\ \vdots \\ \pi_{i,e_{\bar{n}-\bar{d}}} \end{bmatrix} = \begin{bmatrix} \sigma_{i,0}(\mathcal{J}_a) \\ \sigma_{i,1}(\mathcal{J}_a) \\ \vdots \\ \sigma_{i,\bar{r}-1}(\mathcal{J}_a) \end{bmatrix}. \tag{22}$$

Therefore, for any $\mathcal{J}_a \subseteq \mathcal{J}$ and every $i \in \mathcal{I}(\mathcal{J}_a)$, the values $\{\rho_{i,p} \mid p = 1,\dots,\bar{s}-1\}$ and $\{\sum_{g=0}^{u-1} c_{eu+g,i} \mid e \in \mathcal{J}\}$ can be found from the values $\{\sigma_{i,w}(\mathcal{J}_a) \mid w = 0,\dots,\bar{r}-1\}$. It follows that we can recover the values $\{\rho_{i,p} \mid p = 1,\dots,\bar{s}-1\}$ and $\{\sum_{g=0}^{u-1} c_{eu+g,i} \mid e \in \mathcal{J}\}$ for all $i \in \mathcal{I}_a$.

Note that for $i \in \mathcal{I}(\mathcal{J}_a)$, $e \in \mathcal{J}_a \setminus \mathcal{J}_1$, and for $p \neq 0$, we have $i(e,p) \in \mathcal{I}_{a-1}$. By the induction hypothesis, we have recovered the values $\{\sum_{g=0}^{u-1} c_{eu+g,i} \mid i \in \mathcal{I}_{a-1}; e \in \mathcal{J}\setminus\mathcal{J}_1\}$, and therefore, we know the values $\{\sum_{g=0}^{u-1} c_{eu+g,i(e,p)} \mid j \in \mathcal{J}_a \setminus \mathcal{J}_1, p \neq 0\}$ for each $i \in \mathcal{I}_a$. With these values and

$\{\rho(i,p) \mid i \in \mathcal{I}_a, p = 1,\dots,s-1\}$, we can obtain the values $\{\sum_{g=0}^{u-1} c_{e_1\ u+g,i} \mid p = 1,\dots,s-1\}$. Since the values of local nodes $\{c_{e_1\ u+g,i} \mid g \neq g_1\}$ are available, we can further recover the value $c_{j_1,i}$.

Thus, we can obtain the values $\{c_{j_1,i(e_1,p)} \mid p = 0,\dots,\bar{s}-1\}$ and $\{\sum_{g=0}^{u-1} c_{e_1\ u+g,i} \mid e \in \mathcal{J} \setminus \mathcal{J}_1\}$ for every $i \in \mathcal{I}_a$. It follows that we can recover these values for every $i \in \mathcal{I}_a$ and $1 \le a \le \bar{n} - \bar{d}$ from the helper racks $\mathcal{R}$. In conclusion, we can recover the values $\{c_{j_1,i(e_1,p)} \mid i \in \mathcal{I}, p = 0,\dots,\bar{s}-1\} = \{c_{j_1,i} \mid i = 0,\dots,l-1\}$ from the information obtained from the helper racks in $\mathcal{R}$.

Now let us count the number of symbols we access in each helper rack. It is clear from the definition of $\sigma_{i,w}(\mathcal{J}_a)$ that we need to access the symbols $\{c_{eu+g,i} \mid 0 \le g < u, i \in \mathcal{I}\}$ for each $e \in \mathcal{R}$. In other words, we need to access $\bar{s}^{\bar{n}-1} = l/\bar{s}$ symbols on each node in the helper racks; thus, the total number of accessed symbols equals $\bar{d}ul/s$. Moreover, the set of symbols we access in each helper rack depends on index of the host rack but not the index of the helper rack.

Note also that the symbols downloaded to the rack e_1 from any helper rack $e \in \mathcal{R}$ form the subset $\{\sum_{g=0}^{u-1} c_{eu+g,i} \mid i \in \mathcal{I}\}$. Thus, the total amount of information downloaded for the purposes of repair equals

$$\bar{d}|\mathcal{I}| = \bar{d}\bar{s}^{\bar{n}-1} = \frac{\bar{d}l}{\bar{d}-\bar{k}+1}.$$

This is the smallest possible number according to the bound (3), and thus the codes support optimal repair.

II. MDS PROPERTY. We will show that the contents of any $n-r$ nodes suffices to find the values of the remaining r nodes.

Let $\mathcal{K} = \{j_1,\dots,j_r\} \subseteq \{0,\dots,n-1\}$ be the set of r nodes to be recovered from the set of $n-r$ nodes in $[0,n-1]\setminus\mathcal{K}$. Let us write $j_b = e_b u + g_b$ where $0 \le g_b < u-1$ for $b = 1,\dots,r$.

Let $\mathcal{J}$ be the set of distinct $e_b, b = 1\dots,r$. For $1 \le a \le |\mathcal{J}|$, let $\mathcal{J}_a \subseteq \mathcal{J}$ be such that $|\mathcal{J}_a| = a$.

Let $\mathcal{I}_0 = \{i = (i_{\bar{n}-1},\dots,i_0) \in \{0,\dots,l-1\} \mid i_e \neq 0, e \in \mathcal{J}\}$. For $1 \le a \le |\mathcal{J}|$ and $\mathcal{J}_a \subseteq \mathcal{J}$, let $\mathcal{I}(\mathcal{J}_a) = \{i = (i_{\bar{n}-1},\dots,i_0) \in \{0,\dots,l-1\} \mid i_e = 0, e \in \mathcal{J}_a; i_{e'} \neq 0, e' \in \mathcal{J}\setminus\mathcal{J}_a\}$. Let $\mathcal{I}(a) = \bigcup_{\mathcal{J}_a \subseteq \mathcal{J}} \mathcal{I}(\mathcal{J}_a)$ where $1 \le a \le |\mathcal{J}|$. Observe that the sets $I_a, 0 \le a \le |\mathcal{J}|$ partition the set $\{0,1,\dots,l-1\}$.

We will prove by induction that we can recover the nodes in $\mathcal{J}$ from the nodes in $\{0,1,\dots,n-1\}\setminus\mathcal{J}$. First, let us establish the induction basis, i.e., we can recover the values $\{c_{j,i} \mid j \in \mathcal{J}\}$ for every $i \in \mathcal{I}_0$ from the nodes $\{c_j \mid j \in \mathcal{J}^c\}$. From (17), for $i \in \mathcal{I}_0$, we have

$$\sum_{j\in\mathcal{J}} \lambda_j^t c_{j,i} = -\sum_{j\in\mathcal{J}^c} \left(\lambda_j^t c_{j,i} + \delta(i_e) \sum_{p=1}^{\bar{s}-1} \mu_p^t c_{j,i(e,p)}\right). \quad (23)$$

To simplify notation, denote the right-hand side of (23) by $\sigma_{i,t} = \sigma_{i,t}(\emptyset)$. Note that the value of $\sigma_{i,t}$ only depends on the nodes $\{c_j \mid j \in \mathcal{J}^c\}$. Writing (23) in matrix form, we have

$$\begin{bmatrix} 1 & \cdots & 1 \\ \lambda_{j_1} & \cdots & \lambda_{j_r} \\ \vdots & \ddots & \vdots \\ \lambda_{j_1}^{r-1} & \cdots & \lambda_{j_r}^{r-1} \end{bmatrix} \begin{bmatrix} c_{j_1,i} \\ c_{j_2,i} \\ \vdots \\ c_{j_r,i} \end{bmatrix} = \begin{bmatrix} \sigma_{i,0} \\ \sigma_{i,1} \\ \vdots \\ \sigma_{i,r-1} \end{bmatrix}. \quad (24)$$

Therefore, the values $\{c_{j,i} \mid j \in \mathcal{J}\}$ can be calculated from the values $\{\sigma_{i,t} \mid t = 0,\dots,r-1\}$ for every $i \in \mathcal{I}_0$.

Now let us establish the induction step. Suppose we recover the values $\{c_{j,i} \mid j \in \mathcal{J}\}$ for every $i \in \mathcal{I}_{a'}$ and $0 \le a' \le a-1$ from the nodes $\{c_j \mid j \in \mathcal{J}^c\}$, where $1 \le a \le |\mathcal{J}|$.

Now let us fix a set $\mathcal{J}_a \subseteq \mathcal{J}$ and let $i \in \mathcal{I}(\mathcal{J}_a)$. From (17), we have

$$\begin{aligned} \sum_{j\in\mathcal{J}} \lambda_j^t c_{j,i} &= -\sum_{p=1}^{\bar{s}-1} \mu_p^t \sum_{j\in\mathcal{J}:\, e\in\mathcal{J}_a} c_{j,i(e,p)} \\ &\quad - \sum_{j\in\mathcal{J}^c} \left(\lambda_j^t c_{j,i} + \delta(i_e) \sum_{p=1}^{\bar{s}-1} \mu_p^t c_{j,i(e,p)}\right) \\ &=: -\rho'_{i,t} - \sigma_{i,t}(\mathcal{J}_a), \end{aligned} \quad (25)$$

where the last line serves to introduce the shorthand notation. Note that we know the values $\{\sigma_{i,t}(\mathcal{J}_a) \mid t = 0,\dots,r-1\}$ since the value $\sigma_{i,t}(\mathcal{J}_a)$ only depends on the nodes $\{c_j \mid j \in \mathcal{J}^c\}$. Furthermore, we also know the values $\{\rho'_{i,t} \mid t = 0,\dots,r-1\}$. Indeed, for $i \in \mathcal{I}(\mathcal{J}_a)$, $e \in \mathcal{J}_a$, and $p \neq 0$, we have $i(e,p) \in \mathcal{I}_{a-1}$. By the induction hypothesis, we have recovered the values $\{c_{j,i} \mid i \in \mathcal{I}_{a-1}, j \in \mathcal{J}\}$, and therefore, we know the values $\{c_{j,i(e,p)} \mid j \in \mathcal{J}: e \in \mathcal{J}_a, p \neq 0\}$ for each $i \in \mathcal{I}_a$. It follows that we know the values $\{\rho'_{i,t} \mid t = 0,\dots,r-1\}$. Writing (25) in matrix form, we have

$$\begin{bmatrix} 1 & \cdots & 1 \\ \lambda_{j_1} & \cdots & \lambda_{j_r} \\ \vdots & \ddots & \vdots \\ \lambda_{j_1}^{r-1} & \cdots & \lambda_{j_r}^{r-1} \end{bmatrix} \begin{bmatrix} c_{j_1,i} \\ c_{j_2,i} \\ \vdots \\ c_{j_r,i} \end{bmatrix} = \begin{bmatrix} \rho'_{i,0} + \sigma_{i,0}(\mathcal{J}_a) \\ \rho'_{i,1} + \sigma_{i,1}(\mathcal{J}_a) \\ \vdots \\ \rho'_{i,r-1} + \sigma_{i,r-1}(\mathcal{J}_a) \end{bmatrix}. \quad (26)$$

Therefore, the values $\{c_{j,i} \mid j \in \mathcal{J}\}$ can be recovered for every $i \in \mathcal{I}(\mathcal{J}_a)$ and $\mathcal{J}_a \subseteq \mathcal{J}$. It follows that we can recover the values $\{c_{j,i} \mid j \in \mathcal{J}\}$ for every $i \in \mathcal{I}_a$. Thus, all the values $\{c_{j,i} \mid j \in \mathcal{J}, i \in \mathcal{I}_a, 0 \le a \le |\mathcal{J}|\} = \{c_{j,i} \mid j \in \mathcal{J}, i \in \{0,\dots,l-1\}$ can be recovered from the nodes $\{c_j \mid j \in \mathcal{J}^c\}$.

Since $\mathcal{J}$ is arbitrary, we conclude that any $n-r$ nodes can recover the entire codeword, i.e., the code is MDS. ■

V. A Construction of Reed-Solomon Codes With Optimal Repair

In this section we present a family of scalar MDS codes that support optimal repair of a single node from an arbitrary subset of $\bar{d}$ helper racks. We still use the same notation as in the previous parts of the paper. As noted earlier, the construction is a modification of the RS code family in [25]. The new element of the construction is the idea of coupling the code family of [25] and the multiplicative structure that matches the grouping of the nodes into racks. This latter part is similar to the idea of Sec. III.

Let q be a power of a prime, let u be the size of the rack, and suppose that $u|(q-1)$. Let $k = \bar{k}u + v, 0 \le v \le u-1$, $\bar{s} = \bar{d} - \bar{k} + 1$. Let $p_i, i = 1,\dots,\bar{n}$ be distinct primes such that $p_i \equiv 1 \bmod \bar{s}$ and $p_i > u$ for $i = 1,\dots,\bar{n}$; for instance, we can take the *smallest* $\bar{n}$ primes with these properties. For $i = 1,\dots,\bar{n}$ let λ_i be an element of degree p_i over $\mathbb{F}_q$. Let

$$F_i := \mathbb{F}_q(\lambda_j, j \in \{1,\dots,\bar{n}\}\setminus\{i\}),\ i = 1,\dots,\bar{n}$$
$$\mathbb{F} := \mathbb{F}_q(\lambda_1,\dots,\lambda_{\bar{n}}).$$

Let $\mathbb{K}$ be an extension of $\mathbb{F}$ of degree $\bar{s}$ and let $\mu \in \mathbb{K}$ be a generating element of $\mathbb{K}$ over $\mathbb{F}$. Thus, for any $i = 1, \ldots, \bar{n}$ we have the chain of inclusions

$$\mathbb{F}_q \subset F_i \subset \mathbb{K};$$

so $\mathbb{K}$ is the l-th degree extension of $\mathbb{F}_q$, where $l = [\mathbb{K} : \mathbb{F}_q] = \bar{s} \prod_{m=1}^{\bar{n}} p_m$.

Further, let $\lambda \in \mathbb{F}_q$ be an element of multiplicative order u. Consider the set of elements

$$\lambda_{ij} = \lambda_i \lambda^{j-1}, i = 1, \ldots, \bar{n};\ j = 1, \ldots, u.$$

Consider an RS code $\mathcal{C} = RS_{\mathbb{K}}(n, k, \boldsymbol{\Omega})$ where the set of evaluation points $\boldsymbol{\Omega}$ is as follows:

$$\boldsymbol{\Omega} = \bigcup_{i=1}^{\bar{n}} \Omega_i, \text{ where } \Omega_i = \{\lambda_{ij}, j = 1, \ldots, u\}.$$

A codeword of $\mathcal{C}$ has the form $c = (c_1, c_2, \ldots, c_n)$, where the coordinate $c_m, m = (i-1)u + j, 1 \le i \le \bar{n};\ 1 \le j \le u$ corresponds to the evaluation point λ_{ij}.

To describe the repair procedure, we will need the following easy modification of Lemma 1 of [25].

Lemma V.1. *For $i \in \{1, \ldots, \bar{n}\}$, there exists subspace S_i of $\mathbb{K}$ such that*

$$\dim_{F_i} S_i = p_i, \quad S_i + S_i \lambda_i^u + \cdots + S_i \lambda_i^{u(\bar{s}-1)} = \mathbb{K} \tag{27}$$

where $S_i\beta = \{\gamma\beta, \gamma \in S_i\}$ and the operation $+$ is the Minkowski sum of sets, $T_1 + T_2 := \{\gamma_1 + \gamma_2 : \gamma_1 \in T_1, \gamma_2 \in T_2\}$.

Proof. The space S_i is constructed as follows. Define the following vector spaces over F_i:

$$S_i^{(1)} = \mathrm{Span}_{F_i}(\mu^t \lambda_i^{t+e\bar{s}}, t = 0, 1, \ldots, \bar{s}-1;\ e = 0, 1, \ldots, \tfrac{p_i - 1}{\bar{s}} - 1)$$

$$S_i^{(2)} = \mathrm{Span}_{F_i}\Big(\sum_{t=0}^{\bar{s}-1} \mu^t \lambda_i^{p_i - 1}\Big)$$

and take

$$S_i = S_i^{(1)} + S_i^{(2)}.$$

Now the proof of [25, Lemma 1] can be followed step by step, using the fact that $\{1, \lambda_i^u, \ldots, (\lambda_i^u)^{p_i - 1}\}$ forms a basis for $\mathbb{F}$ over F_i, and we do not repeat it here. □

The main result of this section is given in the following proposition.

Proposition V.2. *The code $\mathcal{C}$ supports optimal repair of a single failed node in any rack from any $\bar{d}$ helper racks.*

The proof follows the scheme in [25] which is itself an implementation of the framework for repair of RS codes proposed in [7].

Proof. Let

$$c = ((c_{(i-1)u+j})_{1 \le i \le \bar{n};\, 1 \le j \le u})$$

be a codeword of $\mathcal{C}$. Suppose that $c_{(i^*-1)u+j^*}$ is the failed node, i.e., the index of the host rack is $i^*, 1 \le i^* \le n$, and the index of the failed node in this rack is $j^*, 1 \le j^* \le u$. Denote by $\mathcal{R} \subseteq \{1, \ldots, \bar{n}\} \setminus \{i^*\}, |\mathcal{R}| = \bar{d}$ the set of helper racks. The repair relies on the information downloaded from all the nodes in $\mathcal{R}$ and the functional nodes in the host rack. Define the annihilator polynomial of the set of locators of all the nodes in $\mathcal{R}$:

$$h(x) = \prod_{\substack{i \in \{1, \ldots, \bar{n}\} \setminus (\mathcal{R} \cup \{i^*\}), \\ 1 \le j \le u}} (x - \lambda_{ij}). \tag{28}$$

Let $t = uw$, where $w = 0, \ldots, \bar{s} - 1$. Since

$$\begin{aligned} \deg x^t h(x) &\le (\bar{s} - 1)u + (\bar{n} - \bar{d} - 1)u \\ &= (\bar{r} - 1)u < \bar{r}u - \upsilon = n - k \end{aligned} \tag{29}$$

evaluations of the polynomials $x^t h(x)$ are contained in the dual code $\mathcal{C}^{\perp}$.

Since $\mathcal{C}^{\perp}$ itself is a (generalized) RS code, there is a vector $a = (a_1, \ldots, a_n) \in (\mathbb{K}^*)^n$ such that any codeword of $\mathcal{C}^{\perp}$ has the form $(a_{ij} f(\lambda_{ij}))_{1 \le i \le \bar{n};\, 1 \le j \le u}$, where $f \in \mathbb{K}[x]$ is a polynomial of degree $\le r - 1$. Thus, by (29), the vector $(a_1 \lambda_{11}^t h(\lambda_{11}), \ldots, a_n \lambda_{\bar{n},u}^t h(\lambda_{\bar{n},u})) \in \mathcal{C}^{\perp}$, so the inner product of this vector and the codeword c is zero. In other words, we have

$$\begin{aligned} &\sum_{j=1}^{u} a_{(i^*-1)u+j} \lambda_{i^*j}^t h(\lambda_{i^*j}) c_{(i^*-1)u+j} \\ &\quad = -\sum_{\substack{i=1, \\ i \ne i^*}}^{\bar{n}} \sum_{j=1}^{u} a_{(i-1)u+j} \lambda_{ij}^t h(\lambda_{ij}) c_{(i-1)u+j}. \end{aligned}$$

Let S_{i^*} be the subspace defined in Lemma V.1 and let $\{e_1, \ldots, e_{p_{i^*}}\}$ be a basis of S_{i^*} over F_{i^*}. Then for $m = 1, \ldots, p_{i^*}$, we have

$$\begin{aligned} &\sum_{j=1}^{u} e_m a_{(i^*-1)u+j} \lambda_{i^*j}^t h(\lambda_{i^*j}) c_{(i^*-1)u+j} \\ &\quad = -\sum_{\substack{i=1, \\ i \ne i^*}}^{\bar{n}} \sum_{j=1}^{u} e_m a_{(i-1)u+j} \lambda_{ij}^t h(\lambda_{ij}) c_{(i-1)u+j}. \end{aligned}$$

Evaluating the trace from $\mathbb{K}$ to F_{i^*} on both sides of the last equation, we obtain

$$\begin{aligned} &\mathrm{tr}_{\mathbb{K}/F_{i^*}}\Big(\sum_{j=1}^{u} e_m a_{(i^*-1)u+j} \lambda_{i^*j}^t h(\lambda_{i^*j}) c_{(i^*-1)u+j}\Big) \\ &= -\sum_{\substack{i=1, \\ i \ne i^*}}^{\bar{n}} \sum_{j=1}^{u} \lambda_{ij}^t h(\lambda_{ij}) \mathrm{tr}_{\mathbb{K}/F_{i^*}}(e_m a_{(i-1)u+j} c_{(i-1)u+j}) \\ &= -\sum_{i \in \mathcal{R}} \sum_{j=1}^{u} \lambda_{ij}^t h(\lambda_{ij}) \mathrm{tr}_{\mathbb{K}/F_{i^*}}(e_m a_{(i-1)u+j} c_{(i-1)u+j}), \end{aligned} \tag{30}$$

where we used (28), the fact that $\lambda_{ij} \in F_{i^*}$ for all $i \ne i^*$, and where $t = uw, w = 0, \ldots, \bar{s} - 1$ and $m = 1, \ldots, p_{i^*}$.

Recall that $\lambda_{ij} = \lambda_i \lambda^{j-1}$ and $\lambda^u = 1$. From (30) we have

$$\begin{aligned}&\mathrm{tr}_{\mathbb{K}/F_{i^*}}\left(e_m \lambda_{i^*}^{uw} \sum_{j=1}^{u} a_{(i^*-1)u+j} h(\lambda_{i^*j}) c_{(i^*-1)u+j}\right)\\ &= -\sum_{i\in\mathcal{R}} \lambda_i^{uw} \sum_{j=1}^{u} h(\lambda_{ij}) \mathrm{tr}_{\mathbb{K}/F_{i^*}}(e_m a_{(i-1)u+j} c_{(i-1)u+j}),\end{aligned} \tag{31}$$

where the parameters t, m are as above. By (27) in Lemma V.1 and the definition of the set $\{e_m\}$, the set $\{e_m \lambda_{i^*}^{uw} \mid 1 \le m \le p_{i^*}, 0 \le w \le \bar{s}-1\}$ forms a basis for $\mathbb{K}$ over F_{i^*}. Therefore, the mapping

$$\beta \mapsto \mathrm{tr}_{\mathbb{K}/F_{i^*}}(e_m \lambda_{i^*}^{uw} \beta),\ 1 \le m \le p_{i^*},\ 0 \le w \le \bar{s}-1 \tag{32}$$

is a bijection.

The repair procedure is accomplished as follows. For every $i \in \mathcal{R}$, the elements

$$\sum_{j=1}^{u} h(\lambda_{ij}) \mathrm{tr}_{\mathbb{K}/F_{i^*}}(e_m a_{(i-1)u+j} c_{(i-1)u+j}),\ m = 1, \ldots, p_{i^*} \tag{33}$$

are downloaded from helper rack i. By (31) this enables us to find the elements

$$\mathrm{tr}_{\mathbb{K}/F_{i^*}}\left(e_m \lambda_{i^*}^{uw} \sum_{j=1}^{u} a_{(i^*-1)u+j} h(\lambda_{i^*j}) c_{(i^*-1)u+j}\right),$$

$m = 1, \ldots, p_{i^*}$. Next, on account of (32) we can find the value of $\sum_{j=1}^{u} a_{(i^*-1)u+j} h(\lambda_{i^*j}) c_{(i^*-1)u+j}$. Finally, since the values of the coordinates $c_{(i^*-1)u+j}$, $j \ne j^*$ stored on the functional nodes in the host rack i^* are available and the entires of the vector a are nonzero, we can find $c_{(i^*-1)u+j^*}$, completing the repair.

The number of field symbols of F_{i^*} (33) transmitted from the helper racks to the host rack equals $\bar{d} p_{i^*}$. Therefore, we conclude that the repair bandwidth of $\mathcal{C}$ is

$$\frac{\bar{d} p_{i^*}}{[\mathbb{K} : F_{i^*}]} l = \frac{\bar{d} l}{\bar{s}}. \tag{34}$$

This meets the bound (3) with equality, and proves the claim of optimal repair. □

Appendix A
Proof of Proposition II.3

Let $\mathcal{I} \subset \mathcal{R}$, $|\mathcal{I}| = \bar{k}-1$ be a subset of helper racks. Since $(\bar{d}-\bar{k}+1)u \ge d-k+1$, Lemma II.2 implies that

$$\sum_{i\in\mathcal{R}\setminus\mathcal{I}} \beta_i \ge l. \tag{35}$$

Let us sum the left-hand side on all $\mathcal{I} \subset \mathcal{R}$, $|\mathcal{I}| = \bar{k}-1$:

$$\sum_{\substack{\mathcal{I}\subset\mathcal{R}\\ |\mathcal{I}|=\bar{k}-1}} \sum_{i\in\mathcal{R}\setminus\mathcal{I}} \beta_i = \sum_{i\in\mathcal{R}} \sum_{\substack{\mathcal{I}\subset\mathcal{R}\\ \mathcal{I}\not\ni i}} \beta_i = \binom{\bar{d}-1}{\bar{k}-1} \sum_{i\in\mathcal{R}} \beta_i.$$

Together with (35) we obtain

$$\binom{\bar{d}-1}{\bar{k}-1} \sum_{i_t\in\mathcal{R}} \beta_i \ge \binom{\bar{d}}{\bar{k}-1} l$$

or

$$\sum_{i\in\mathcal{R}} \beta_i \ge \frac{\bar{d} l}{\bar{d}-\bar{k}+1},$$

i.e., (3). Moveover, this bound holds with equality if and only if (35) holds with equality for every $\mathcal{I} \subset \mathcal{R}$, $|\mathcal{I}| = \bar{k}-1$. Suppose for the sake of contradiction that the uniform download claim does not hold, and there is a rack i such that $\beta_i \ne l/\bar{s}$, for instance, $\beta_i < l/\bar{s}$, where $\bar{s} = \bar{d}-\bar{k}+1$. Let $\mathcal{J} \subset \mathcal{R}$, $|\mathcal{J}| = \bar{s}$, $i \in \mathcal{J}$. There must be a rack $i_1 \in \mathcal{J}$ that contributes more than the average number of symbols, i.e., $\beta_{i_1} > l/\bar{s}$. Consider the subset $(\mathcal{J}\setminus\{i\})\cup\{i_2\}$, where $i_2 \ne i$ is another element of $\mathcal{R}$ (which exists since $\bar{k} > 1$ implies $|\mathcal{J}| < |\mathcal{R}|$). We have that $\beta_{i_2} < l/\bar{s}$. Now take the subset $I = (\mathcal{J}\setminus\{i_1\}) \cup \{i_2\}$ and note that for it, (35) fails to hold with equality, a contradiction.

Appendix B
Proof of Prop. II.6

Proof: Let $m' \in \mathcal{R}$ and let $\mathcal{I}$ be a subset of $u-v$ nodes in rack m', where $0 \le v \le u-1$. Let $\mathcal{J} \subseteq \mathcal{R}\setminus\{m'\}$, $|\mathcal{J}| = \bar{d}-\bar{k}$ be a subset of helper racks. These racks contain $u(\bar{d}-\bar{k}) = d-k+1-(u-v)$ nodes in these racks, and together with the nodes in the set $\mathcal{I}$ this forms a group of $d-k+1$ nodes. Using Lemma II.2, we have

$$\sum_{m\in\mathcal{J}} \sum_{e=1}^{u} \alpha_{m,e} + \sum_{e\in\mathcal{I}} \alpha_{m',e} \ge l. \tag{36}$$

Let us average over the $\binom{u}{u-v}$ choices of the set $\mathcal{I}$:

$$\binom{u}{u-v} \sum_{m\in\mathcal{J}} \sum_{e=1}^{u} \alpha_{m,e} + \sum_{\mathcal{I}:|\mathcal{I}|=u-v} \sum_{e\in\mathcal{I}} \alpha_{m',e} \ge \binom{u}{u-v} l.$$

Interchanging the sums in the second term on the left, we obtain

$$\begin{aligned}\binom{u}{u-v} \sum_{m\in\mathcal{J}} \sum_{e=1}^{u} \alpha_{m,e} + \binom{u-1}{u-v-1} \sum_{e=1}^{u} \alpha_{m',e}\\ \ge \binom{u}{u-v} l.\end{aligned}$$

or

$$\frac{u}{u-v} \sum_{m\in\mathcal{J}} \sum_{e=1}^{u} \alpha_{m,q} + \sum_{e=1}^{u} \alpha_{m',e} \ge \frac{u}{u-v} l. \tag{37}$$

Now let us average over the choice of $\mathcal{J} \subset \mathcal{R}\setminus\{m'\}$. Noting that

$$\sum_{\substack{\mathcal{J}\subset\mathcal{R}\setminus\{m'\}\\ |\mathcal{J}|=\bar{s}-1}} \sum_{m\in\mathcal{J}} \sum_{e=1}^{u} \alpha_{m,e} = \binom{\bar{d}-2}{\bar{s}-2} \sum_{m\in\mathcal{R}\setminus\{m'\}} \sum_{e=1}^{u} \alpha_{m,e}$$

we obtain from (37)

$$\begin{aligned}\frac{u}{u-v}\binom{\bar{d}-2}{\bar{s}-2} \sum_{m\in\mathcal{R}\setminus\{m'\}} \sum_{e=1}^{u} \alpha_{m,e}\\ + \binom{\bar{d}-1}{\bar{s}-1} \sum_{e=1}^{u} \alpha_{m',e} \ge \binom{\bar{d}-1}{\bar{s}-1} \frac{u}{u-v} l.\end{aligned}$$

On account of the assumption that $\bar{d} \geq 2, \bar{s} \geq 2$ we find

$$\frac{u}{u-v} \sum_{m \in \mathcal{R} \setminus \{m'\}} \sum_{e=1}^{u} \alpha_{m,e} + \frac{\bar{d}-1}{\bar{s}-1} \sum_{e=1}^{u} \alpha_{m',e} \geq \frac{\bar{d}-1}{\bar{s}-1} \frac{u}{u-v} l. \quad (38)$$

Now let us average on the choice of $m' \in \mathcal{R}$:

$$\frac{u(\bar{d}-1)}{u-v} \sum_{m \in \mathcal{R}} \sum_{e=1}^{u} \alpha_{m,e} + \frac{\bar{d}-1}{\bar{s}-1} \sum_{m' \in \mathcal{R}} \sum_{e=1}^{u} \alpha_{m',e} \geq \frac{\bar{d}-1}{\bar{s}-1} \frac{u}{u-v} \bar{d} l$$

or

$$\alpha := \sum_{m \in \mathcal{R}} \sum_{e=1}^{u} \alpha_{m,e} \geq \frac{\bar{d} u l}{u(\bar{s}-1)+u-v} = \frac{\bar{d} u l}{s}.$$

Equality holds if and only if it holds in (36). This implies the uniform access condition, which is proved in exactly the same way as the uniform download condition in Prop. II.3. ■

Appendix C
Proof of Theorem II.7

Proof of Part (a): The proof will be given for the repair of a node in a systematic rack. In the end we will argue that the claimed bound also applies to the repair of nodes in parity racks.

Without loss of generality, assume $\{\bar{k}+1, \ldots, \bar{k}+\bar{s}\}$ to be the $\bar{s}$ parity racks that are involved in the repair of the failed node. Let $\bar{k}+i, i = 1, \ldots, \bar{s}$ be a helper rack. Since the repair scheme is linear, the information that this rack provides is obtained through a linear transformation of its contents. Denote the matrix of this transformation by $S_{\bar{k}+i,m_1}$ and call it the *repair matrix* for repairing a failed node in rack m_1 from rack $\bar{k}+i$ (and call its row space the *repair subspace* of the node. Note that it is an $\frac{l}{\bar{s}} \times ul$ matrix over F; moreover, for optimal repair, the rank of $S_{\bar{k}+i,m_1}$ necessarily is $l/\bar{s}$ for all $i \in [\bar{s}]$. The information that parity rack $\bar{k}+i$ transmits to repair the failed node in rack m_1 is given by

$$\begin{aligned} S_{\bar{k}+i,m_1} \boldsymbol{c}_{\bar{k}+i} &= S_{\bar{k}+i,m_1} \left(c_{k+(i-1)u+1}, \ldots, c_{k+iu} \right)^T \\ &= S_{\bar{k}+i,m_1} \Big(\sum_{j=1}^{k} A_{(i-1)u+1,j} c_j, \ldots, \sum_{j=1}^{k} A_{iu,j} c_j \Big)^T, \\ & \qquad i = 1, \ldots, \bar{s}. \end{aligned} \quad (39)$$

For given i, j let us define the block matrix $\mathcal{A}_{i,j} = (A_{(i-1)u+1,j}, \ldots, A_{iu,j})^T$ (a part of column j in the encoding matrix that corresponds to rack m). Suppose that the index of the failed node in rack m_1 is j_1, and note that $(m_1-1)u+1 \leq j_1 \leq m_1 u$. Then from (39) we obtain

$$S_{\bar{k}+i,m_1} \boldsymbol{c}_{\bar{k}+i} = S_{\bar{k}+i,m_1} \mathcal{A}_{i,j_1} c_{j_1} + S_{\bar{k}+i,m_1} \sum_{\substack{j=1 \\ j \neq j_1}}^{k} \mathcal{A}_{i,j_1} c_j, \quad i = 1, \ldots, \bar{s}. \quad (40)$$

From (40) we observe that the information that parity rack $\bar{k}+i$ provides for the repair of node c_{j_1} contains interference from the other systematic nodes c_j, $j \neq j_1$. Moreover, as rack m_1 collects all the information sent from the helper racks $\bar{k}+i$, $1 \leq i \leq \bar{s}$, in order to repair node c_j, it is necessary that

$$\operatorname{rank} \begin{bmatrix} S_{\bar{k}+1,m_1} \mathcal{A}_{1,j_1} \\ \vdots \\ S_{\bar{k}+\bar{s},m_1} \mathcal{A}_{\bar{s},j_1} \end{bmatrix} = l. \quad (41)$$

This relation holds true because Equations (40) evaluate a linear combination of the contents of the nodes in the host rack. To retrieve the l symbols of the failed node from this linear combination, condition (41) is necessary.

Let us further define the $ul \times ul$ matrix $\mathcal{D}_{i,m} = (\mathcal{A}_{i,(m-1)u+1}, \ldots, \mathcal{A}_{i,mu})$ by assembling together u columns of the form $\mathcal{A}_{\cdot,\cdot}$, i.e., $\mathcal{D}_{i,m} = (A_{\alpha,\beta})$, $(i-1)u+1 \leq \alpha \leq iu$, $(m-1)u+1 \leq \beta \leq mu$. These matrices are defined for notational convenience and enable us to argue about entire racks rather than their elements. Since the code $\mathcal{C}$ is MDS, the matrix $\mathcal{D}_{i,m}$ is invertible for all $1 \leq i \leq \bar{r}$ and $1 \leq m \leq \bar{k}$. Rewriting (40) with this notation, we obtain

$$S_{\bar{k}+i,m_1} \boldsymbol{c}_{\bar{k}+i} = S_{\bar{k}+i,m_1} \mathcal{D}_{i,m_1} \boldsymbol{c}_{m_1} + S_{\bar{k}+i,m_1} \sum_{\substack{m=1 \\ m \neq m_1}}^{\bar{k}} \mathcal{D}_{i,m} \boldsymbol{c}_m. \quad (42)$$

Since $(m_1-1)u+1 \leq j_1 \leq m_1 u$, from (41) we have

$$\operatorname{rank} \begin{bmatrix} S_{\bar{k}+1,m_1} \mathcal{D}_{1,m_1} \\ \vdots \\ S_{\bar{k}+\bar{s},m_1} \mathcal{D}_{\bar{s},m_1} \end{bmatrix} = l. \quad (43)$$

So far we have only considered the information provided by the parity racks. It remains to characterize the information transmitted by the systematic racks. From (42), in order to cancel out the interference from systematic rack $m \neq m_1$, rack m_1 needs to download from systematic rack m the vector $(S_{\bar{k}+1,m_1} \mathcal{D}_{1,m} \boldsymbol{c}_m, \ldots, S_{\bar{k}+\bar{s},m_1} \mathcal{D}_{\bar{s},m} \boldsymbol{c}_m)^T$. By Proposition II.3, for optimal repair we necessarily have

$$\operatorname{rank} \begin{bmatrix} S_{\bar{k}+1,m_1} \mathcal{D}_{1,m} \\ \vdots \\ S_{\bar{k}+\bar{s},m_1} \mathcal{D}_{\bar{s},m} \end{bmatrix} = \frac{l}{\bar{s}}. \quad (44)$$

The rank conditions (43) and (44) give rise to the following subspace conditions. For any $m_1 \in [\bar{k}]$,

$$\bigoplus_{i=1}^{\bar{s}} \langle S_{\bar{k}+i,m_1} \mathcal{D}_{i,m_1} \rangle = F^l, \quad (45)$$

$$\langle S_{\bar{k}+1,m_1} \mathcal{D}_{1,m} \rangle = \ldots = \langle S_{\bar{k}+\bar{s},m_1} \mathcal{D}_{\bar{s},m} \rangle, \quad m \neq m_1. \quad (46)$$

The proof of the lower bounds in the theorem relies on these necessary conditions. Let us first bound the dimension of the intersection of the row spaces of the repair matrices.

Lemma C.1. *Let $\mathcal{J} \subset [\bar{k}]$ be a subset of systematic nodes such that $|\mathcal{J}| \leq k-1$ and $m_1 \in \mathcal{J}$. Then for any $i, i' \geq 1$*

$$\dim \bigcap_{m \in \mathcal{J}} \langle S_{\bar{k}+i,m} \rangle = \dim \bigcap_{m \in \mathcal{J}} \langle S_{\bar{k}+i',m} \rangle. \quad (47)$$

Proof: Let $a \in [\bar{k}] \setminus \mathcal{J}$. Since $\mathcal{D}_{i,a}$ is nonsingular, for each $i \in [\bar{s}]$ we have

$$\bigcap_{m\in\mathcal{J}} \langle S_{\bar{k}+i,m}\mathcal{D}_{i,a}\rangle = \Big(\bigcap_{m\in\mathcal{J}} \langle S_{\bar{k}+i,m}\rangle\Big)\mathcal{D}_{i,a}.$$

On the previous line we take the intersection of the subspaces as indicated, then write a basis of the resulting subspace into rows of a matrix, which has ul columns. Since this is also the number of rows of $\mathcal{D}_{i,a}$, this operation is well defined.

Since $a \notin \mathcal{J}$, for any $i, i' \in [\bar{s}]$, from (46) we have

$$\bigcap_{m\in\mathcal{J}} \langle S_{\bar{k}+i,m}\mathcal{D}_{i,a}\rangle = \bigcap_{m\in\mathcal{J}} \langle S_{\bar{k}+i',m}\mathcal{D}_{i',a}\rangle.$$

Therefore, for any $i, i' \in [\bar{s}]$

$$\Big(\bigcap_{m\in\mathcal{J}} \langle S_{\bar{k}+i,m}\rangle\Big)\mathcal{D}_{i,a} = \Big(\bigcap_{m\in\mathcal{J}} \langle S_{\bar{k}+i',m}\rangle\Big)\mathcal{D}_{i',a}. \tag{48}$$

Since $\mathcal{D}_{i,a}$ and $\mathcal{D}_{i',a}$ are invertible, (47) follows. ∎

Now consider the subspace $\bigcap_{m\in\mathcal{J}}\langle S_{\bar{k}+i,m}\mathcal{D}_{i,m_1}\rangle$, where $\mathcal{J}$ is as in Lemma C.1. For any $i' \in [\bar{s}]$, we have

$$\begin{aligned}
&\bigcap_{m\in\mathcal{J}} \langle S_{\bar{k}+i,m}\mathcal{D}_{i,m_1}\rangle \\
&= \langle S_{\bar{k}+i,m_1}\mathcal{D}_{i,m_1}\rangle \bigcap \Big(\bigcap_{m\in\mathcal{J}\setminus\{m_1\}} \langle S_{\bar{k}+i,m}\mathcal{D}_{i,m_1}\rangle\Big) \\
&= \langle S_{\bar{k}+i,m_1}\mathcal{D}_{i,m_1}\rangle \bigcap \Big(\bigcap_{m\in\mathcal{J}\setminus\{m_1\}} \langle S_{\bar{k}+i',m}\mathcal{D}_{i',m_1}\rangle\Big) \\
&\subseteq \bigcap_{m\in\mathcal{J}\setminus\{m_1\}} \langle S_{\bar{k}+i',m}\mathcal{D}_{i',m_1}\rangle.
\end{aligned} \tag{49}$$

Summing on $i \in [\bar{s}]$ on both sides of (49), we obtain

$$\bigoplus_{i=1}^{\bar{s}} \Big(\bigcap_{m\in\mathcal{J}} \langle S_{\bar{k}+i,m}\mathcal{D}_{i,m_1}\rangle\Big) \subseteq \bigcap_{m\in\mathcal{J}\setminus\{m_1\}} \langle S_{\bar{k}+i',m}\mathcal{D}_{i',m_1}\rangle. \tag{50}$$

Note that this is a direct sum because the subspaces $\bigcap_{m\in\mathcal{J}}\langle S_{\bar{k}+i,m}\mathcal{D}_{i,m_1}\rangle$ form a subset of the set of subspaces on the left-hand side of (45) and therefore (for different i) are disjoint.

Since $\mathcal{D}_{i,m_1}$ and $\mathcal{D}_{i',m_1}$ are invertible, we conclude that

$$\sum_{i=1}^{\bar{s}} \dim \bigcap_{m\in\mathcal{J}} \langle S_{\bar{k}+i,m}\rangle \le \dim \bigcap_{m\in\mathcal{J}\setminus\{m_1\}} \langle S_{\bar{k}+i',m}\rangle.$$

Note that from (47), we have

$$\sum_{i=1}^{\bar{s}} \dim \bigcap_{m\in\mathcal{J}} \langle S_{\bar{k}+i,m}\rangle = \bar{s} \dim \bigcap_{m\in\mathcal{J}} \langle S_{\bar{k}+i,m}\rangle.$$

Therefore,

$$\begin{aligned}
\dim \bigcap_{m\in\mathcal{J}} \langle S_{\bar{k}+i,m}\rangle &\le \frac{1}{\bar{s}} \dim \bigcap_{m\in\mathcal{J}\setminus\{m_1\}} \langle S_{\bar{k}+i',m}\rangle \\
&\le \frac{1}{\bar{s}^{|\mathcal{J}|-1}} \dim\langle S_{\bar{k}+i',m}\rangle \\
&\le \frac{l}{\bar{s}^{|\mathcal{J}|}}.
\end{aligned} \tag{51}$$

If $l \ge \bar{s}^{\bar{k}-1}$, then (7) is proved, so let us assume that $l < \bar{s}^{\bar{k}-1}$.

Lemma C.2. *Let $\mathcal{T} \subset [\bar{n}]\setminus\{\bar{k}+i\}$ be a subset such that*

- $1 \le |\mathcal{T}| \le \bar{n}-1$,
- $m_1 \in \mathcal{T}$,
- $\mathcal{T}$ *contains* $\min(\bar{k}-1, |\mathcal{T}|)$ *systematic racks.*

Then

$$\dim \bigcap_{m\in\mathcal{T}} \langle S_{\bar{k}+i,m}\rangle \le \frac{l}{\bar{s}^{|\mathcal{T}|}}. \tag{52}$$

Remark: Some of the repair matrices in (52) refer to the repair scheme of a parity node (a node in a parity rack) using information from another parity rack. These matrices exist and are well defined because by assumption, the code $\mathcal{C}$ supports optimal repair of any node from any set of $\bar{d}$ helper racks.

Proof. By the assumption before the lemma, Eq. (51) holds for any $\mathcal{J}$ of size $\le \bar{k}-1$, which proves the claim for the case $|\mathcal{T}| \le \bar{k}-1$. At the same time, if $|\mathcal{T}| > \bar{k}-1$, take $|\mathcal{J}| = \bar{k}-1$ in (51) and note that $\mathcal{J} \subset \mathcal{T}$. In this case (51) implies (52) and the proof is complete. □

From (52) we observe that the subspaces $\langle S_{\bar{k}+i,m}\rangle, m \in \mathcal{T}$ have a vector in common if and only if $|\mathcal{T}| \le \log_{\bar{s}} l$. Now consider a $ul \times (\bar{n}-1)$ matrix V whose rows correspond to the ul vectors in the standard basis of F^{ul} and columns to the repair matrices $S_{\bar{k}+i,m}, m \in [\bar{n}] \setminus \{\bar{k}+i\}$. Put $V_{im} = 1$ if the ith vector is one of the rows of the mth repair matrix and 0 otherwise. The code has the optimal access property if and only if the rows of the repair matrices are formed of standard basis vectors. Every column of V contains $l/\bar{s}$ ones, and if $|\mathcal{T}| \le \log_{\bar{s}} l$, then every row contains at most $\log_{\bar{s}} l$ ones; thus

$$\frac{l}{\bar{s}}(\bar{n}-1) \le ul \log_{\bar{s}} l.$$

It follows that

$$l \ge \bar{s}^{\frac{\bar{n}-1}{s}}, \tag{53}$$

where we used $s = \bar{s}u$. This concludes the proof of Part (a).

Proof of Part (b): We closely follow the arguments in Part (a) with the only difference that the set $\mathcal{T}$ can now be of size $\bar{n}$, which is possible because the repair matrices are independent of the choice of the helper racks.

Let us outline the argument. Let $|\mathcal{J}| = \bar{k}-1$ and $l < \bar{s}^{\bar{k}-1}$. In this case (51) implies that

$$\dim \bigcap_{m\in\mathcal{J}} \langle S_m\rangle = 0$$

(even in the case when the repair scheme is chosen based on the location of the helpers, and all the more so in the current case). It follows that, for any $\mathcal{T} \subseteq [\bar{n}]$ such that $\mathcal{T} \supseteq \mathcal{J}$, we have

$$\dim \bigcap_{m\in\mathcal{T}} S_m = 0.$$

Therefore, for $l < \bar{s}^{\bar{k}-1}$, we have

$$\dim \bigcap_{m\in\mathcal{T}} S_m \le \frac{l}{\bar{s}^{|\mathcal{T}|}}, \tag{54}$$

for $1 \le |\mathcal{T}| \le \bar{n}$. For the left-hand side of the above inequality to be greater than 1, we necessarily have $|\mathcal{T}| \le \log_{\bar{s}} l$.

Repeating the argument that led to (53), we obtain

$$l \ge \bar{s}^{\bar{n}/s}. \tag{55}$$

Thus, we have proved cases (a) and (b) of the theorem for repairing a failed node in a systematic rack. The same bounds hold for repairing a failed node in any of the parity racks. Indeed, note that a parity rack $\boldsymbol{c}_{\bar{k}+i}, i = 1, \ldots, \bar{r}$ is computed from the systematic racks as follows:

$$\boldsymbol{c}_{\bar{k}+i} = \sum_{j=1}^{\bar{k}} \mathcal{D}_{\bar{k}+i,j} \boldsymbol{c}_j \tag{56}$$

If a node in rack $\bar{k}+i$ has failed, we first choose $\bar{d}$ helper racks and isolate any $(\bar{k}-1)$-subset of the chosen $\bar{d}$-set. Then we write equations of the form (56) where on the right we use these $\bar{k}-1$ racks together with the host rack to express the code symbols in the remaining $\bar{r}$ racks. These equations are obtained from (56) using obvious matrix transformations (no complications arise because the code $\mathcal{C}$ is MDS). After that, we can repeat the proofs given above, which establishes our claim.

REFERENCES

[1] S. Akhlaghi, A. Kiani, and M. R. Ghanavati, "Cost-bandwidth tradeoff in distributed storage systems," *Comput. Commun.*, vol. 33, no. 17, pp. 2105–2115, 2010.

[2] S. B. Balaji, M. N. Krishnan, M. Vajha, V. Ramkumar, B. Sasidharan, and P. V. Kumar, "Erasure coding for distributed storage: An overview," *Sci. China Inf. Sci.*, vol. 61, Oct. 2018, Art. no. 100301.

[3] S. B. Balaji and P. V. Kumar, "A tight lower bound on the sub-packetization level of optimal-access MSR and MDS codes," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Vail, CO, USA, Jun. 2018, pp. 2381–2385.

[4] V. R. Cadambe, S. A. 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] A. G. Dimakis, P. B. Godfrey, Y. Wu, M. J. Wainwright, and K. Ramchandran, "Network coding for distributed storage systems," *IEEE Trans. Inf. Theory*, vol. 56, no. 9, pp. 4539–4551, Sep. 2010.

[6] B. Gastón, J. Pujol, and M. Villanueva, "A realistic distributed storage system that minimizes data storage and repair bandwidth," in *Proc. Data Compress. Conf.*, Mar. 2013, p. 491.

[7] V. Guruswami and M. Wootters, "Repairing Reed–Solomon codes," *IEEE Trans. Inf. Theory*, vol. 63, no. 9, pp. 5684–5698, Sep. 2017.

[8] H. Hou, P. P. C. Lee, K. W. Shum, and Y. Hu, "Rack-aware regenerating codes for data centers," *IEEE Trans. Inf. Theory*, vol. 65, no. 8, pp. 4730–4745, Aug. 2019.

[9] Y. Hu, P. P. C. Lee, and X. Zhang, "Double regenerating codes for hierarchical data centers," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jul. 2016, pp. 245–249.

[10] Y. Hu *et al.*, "Optimal repair layering for erasure-coded data centers: From theory to practice," *ACM Trans. Storage*, vol. 13, no. 4, 2017, Art. no. 33.

[11] A.-M. Kermarrec, N. Le Scouarnec, and G. Straub, "Repairing multiple failures with coordinated and adaptive regenerating codes," in *Proc. IEEE Int. Symp. Netw. Coding (NetCod)*, Jul. 2011, pp. 1–6.

[12] J. Li and B. Li, "Cooperative repair with minimum-storage regenerating codes for distributed storage," in *Proc. IEEE INFOCOM*, Apr./May 2014, pp. 316–324.

[13] J. Li, X. Tang, and C. Tian, "A generic transformation to enable optimal repair in MDS codes for distributed storage systems," *IEEE Trans. Inf. Theory*, vol. 64, no. 9, pp. 6257–6267, Sep. 2018.

[14] J. Pernas, C. Yuen, B. Gastón, and J. Pujol, "Non-homogeneous two-rack model for distributed storage systems," in *Proc. IEEE Int. Symp. Inf. Theory*, Jul. 2013, pp. 1237–1241.

[15] N. Prakash, V. Abdrashitov, and M. Médard, "The storage versus repair-bandwidth trade-off for clustered storage systems," *IEEE Trans. Inf. Theory*, vol. 64, no. 8, pp. 5783–5805, Aug. 2018.

[16] K. V. Rashmi, N. B. Shah, and P. V. Kumar, "Optimal exact-regenerating codes for distributed storage at the MSR and MBR points via a product-matrix construction," *IEEE Trans. Inf. Theory*, vol. 57, no. 8, pp. 5227–5239, Aug. 2011.

[17] A. S. Rawat, O. O. Koyluoglu, and S. Vishwanath, "Centralized repair of multiple node failures with applications to communication efficient secret sharing," *IEEE Trans. Inf. Theory*, vol. 64, no. 12, pp. 7529–7550, Dec. 2018.

[18] S. Sahraei and M. Gastpar, "Increasing availability in distributed storage systems via clustering," 2017, *arXiv:1710.02653v2*. [Online]. Available: https://arxiv.org/abs/1710.02653

[19] B. Sasidharan, M. Vajha, and P. V. Kumar, "An explicit, coupled-layer construction of a high-rate MSR code with low sub-packetization level, small field size and all-node repair," 2016, *arXiv:1607.07335*. [Online]. Available: https://arxiv.org/abs/1607.07335

[20] K. W. Shum and Y. Hu, "Cooperative regenerating codes," *IEEE Trans. Inf. Theory*, vol. 59, no. 11, pp. 7229–7258, Nov. 2013.

[21] J.-Y. Sohn, B. Choi, and J. Moon, "A class of MSR codes for clustered distributed storage," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jun. 2018, pp. 2366–2370.

[22] J.-Y. Sohn, B. Choi, S. W. Yoon, and J. Moon, "Capacity of clustered distributed storage," *IEEE Trans. Inf. Theory*, vol. 65, no. 1, pp. 81–107, Jan. 2019.

[23] 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.

[24] I. Tamo, Z. Wang, and J. Bruck, "Access versus bandwidth in codes for storage," *IEEE Trans. Inf. Theory*, vol. 60, no. 4, pp. 2028–2037, Apr. 2014.

[25] I. Tamo, M. Ye, and A. Barg, "The repair problem for Reed–Solomon codes: Optimal repair of single and multiple erasures with almost optimal node size," *IEEE Trans. Inf. Theory*, vol. 65, no. 5, pp. 2673–2695, May 2019.

[26] M. A. Tebbi, T. H. Chan, and C. W. Sung, "A code design framework for multi-rack distributed storage," in *Proc. IEEE Inf. Theory Workshop (ITW)*, Nov. 2014, pp. 55–59.

[27] M. Vajha *et al.*, "Clay codes: Moulding MDS codes to yield an MSR code," in *Proc. 16th USENIX Conf. File Storage Technol. (FAST)*, Oakland, CA, USA, Feb. 2018, pp. 139–154.

[28] M. Ye and A. Barg, "Explicit constructions of high-rate MDS array codes with optimal repair bandwidth," *IEEE Trans. Inf. Theory*, vol. 63, no. 4, pp. 2001–2014, Apr. 2017.

[29] M. Ye and A. Barg, "Cooperative repair: Constructions of optimal MDS codes for all admissible parameters," *IEEE Trans. Inf. Theory*, vol. 65, no. 3, pp. 1639–1656, Mar. 2019.

[30] M. Ye and A. Barg, "Explicit constructions of optimal-access MDS codes with nearly optimal sub-packetization," *IEEE Trans. Inf. Theory*, vol. 63, no. 10, pp. 6307–6317, Oct. 2017.

Zitan Chen received his B.Eng. degree in Information Engineering in 2015 from the Chinese University of Hong Kong, Hong Kong. He is currently working toward his Ph.D. degree in the Department of Electrical and Computer Engineering and Institute for Systems Research, University of Maryland, College Park.

Alexander Barg (M'00–SM'01–F'08) is a professor in the Department of Electrical and Computer Engineering and Institute for Systems Research, University of Maryland, College Park, MD. He is broadly interested in information and coding theory, applied probability, and algebraic combinatorics, and has published about a hundred research papers. He received the 2015 Information Theory Society paper award, was a plenary speaker at the 2016 IEEE International Symposium on Information Theory (Barcelona, Spain), and currently serves Editor-in-Chief of this TRANSACTIONS.