Cooperative repair: Constructions of optimal MDS codes for all admissible parameters

Min Ye Alexander Barg, *Fellow, IEEE*

***Abstract*—Two widely studied models of multiple-node repair in distributed storage systems are *centralized repair* and *cooperative repair*. The centralized model assumes that all the failed nodes are recreated in one location, while the cooperative one stipulates that the failed nodes may communicate but are distinct, and the amount of data exchanged between them is included in the repair bandwidth.**

As our first result, we prove a lower bound on the minimum bandwidth of cooperative repair. We also show that the cooperative model is stronger than the centralized one, in the sense that any MDS code with optimal repair bandwidth under the former model also has optimal bandwidth under the latter one. These results were previously known under the additional "uniform download" assumption, which is removed in our proofs.

As our main result, we give explicit constructions of MDS codes with optimal cooperative repair for all possible parameters. More precisely, given any n, k, h, d such that $2 \leqslant h \leqslant n-d \leqslant n-k$ we construct (n, k) MDS codes over the field F of size $|F| \geqslant (d+1-k)n$ that can optimally repair any h erasures from any d helper nodes. The repair scheme of our codes involves two rounds of communication. In the first round, each failed node downloads information from the helper nodes, and in the second one, each failed node downloads additional information from the other failed nodes. This implies that our codes achieve the optimal repair bandwidth using the smallest possible number of rounds.

***Index Terms*—Distributed storage, MDS codes, MSR codes, Multiple-node repair, Regenerating codes.**

I. Introduction

A. Centralized and cooperative repair models

The problem considered in this paper is motivated by the distributed nature of the system wherein the coded data is distributed across a large number of physical storage nodes. When some storage nodes fail, the repair task performed by the system relies on communication between individual nodes, which introduces new challenges in the code design. Coding schemes that address these challenges are known under the name of *regenerating codes*, a concept that was isolated and studied in the work of Dimakis et. al. [1]. In paper [1] the authors suggested a new metric that has a bearing on the overall efficiency of the system, namely, the *repair bandwidth*, i.e., the amount of data communicated between the nodes in the process of repairing failed nodes. Most works on this class of codes assume that the information is protected with Maximum Distance Separable (MDS) codes which provide the optimal tradeoff between failure tolerance and storage overhead. Paper [1] also gave a lower bound on the minimum repair bandwidth of MDS codes, known as the *cut-set bound*. Code families that achieve this bound with equality are said to have the *optimal repair property*. Constructions of optimal-repair MDS codes (also known as *minimum storage regenerating*, or MSR codes) were proposed in [2]–[7].

To encode information with an MDS code, the original file is divided into k information blocks viewed as vectors over a finite field F. The encoding procedure then finds $r = n-k$ parity blocks, also viewed as vectors over F, which together with the information blocks form a codeword of a code of length n. The n blocks of the codeword are stored on n different storage nodes. Motivated by this model, we also refer to the coordinates of the codeword as nodes. The task of node repair therefore amounts to erasure correction with the chosen code, and the special feature of the erasure correction problem arising from the distributed data placement is the constraint on the repair bandwidth involved in the repair procedure.

Most studies of MDS codes with optimal repair bandwidth in the literature are concerned with a particular subclass of codes known as MDS *array codes* [8]. An (n, k, l) MDS array code over a finite field F is formed of k information nodes and $r = n-k$ parity nodes with the property that the contents of any k out of n nodes suffices to recover the codeword. Every node is a column vector in F^l, reflecting the fact that the system views a large data block stored in one node as one coordinate of the codeword. The parameter l that determines the dimension of each node is called *sub-packetization*.

While originally the repair problem was confined to a single node failure, studies into regenerating codes have expanded into the task of repairing multiple erasures. The problem of repairing multiple erasures comes in two variations. One of them is the *centralized model*, where a single data center is responsible for the repair of all the failed nodes [4], [9]–[14], and the other is the *cooperative model*, where the failed nodes may communicate but are distinct, and the amount of data exchanged between them is included in the repair bandwidth [15]–[18]. The cut-set bounds on the repair bandwidth for multiple erasures under these two models were derived in [9] and [16] respectively.

Let $\mathcal{F} \subset [n], |\mathcal{F}| = h$ and $\mathcal{R} \subseteq [n]\backslash\mathcal{F}, |\mathcal{R}| = d$ be the sets of indices of the failed nodes and the helper nodes, respectively, where we use the notation $[n] := \{1, 2, \ldots, n\}$. Informally speaking, under the centralized model, repair proceeds by downloading $\beta_j, j \in \mathcal{R}$ symbols of F from each of the helper nodes $C_j, j \in \mathcal{R}$, and computing the values of the failed nodes. It is assumed that the repair is performed by a

The results of this paper were presented in part at the International Symposium on Information Theory (ISIT, Vail, CO, June 2018).

M. Ye is with Department of Electrical Engineering, Princeton University, Princeton, NJ, email: yeemmi@gmail.com.

A. Barg is with Dept. of ECE and ISR, University of Maryland, College Park, MD 20742, and IITP, Russian Academy of Sciences, 127051 Moscow, Russia. Email: abarg@umd.edu. Research supported in part by NSF grants CCF 1422955 and CCF 1618603.

data center having access to all the downloaded information, and so the *repair bandwidth* equals $\beta_{\mathcal{F}}(\mathcal{R}) = \sum_{j\in\mathcal{R}} \beta_j$. The variation introduced by the cooperative model does not include the data center, and so the repair bandwidth includes not only the information downloaded from the helper nodes but also the information exchanged between the failed nodes in the repair process. In other words, under the centralized model, each failed node has access to all the data downloaded from the helper nodes, while under the cooperative model, each failed node only has access to its own downloaded data.

B. Formal statement of the problems

Consider an (n, k, l) MDS array code $\mathcal{C}$ over a finite field F and let $C \in \mathcal{C}$ be a codeword. We write C as $(C_1, C_2, \dots, C_n)$, where $C_i = (c_{i,0}, c_{i,1}, \dots, c_{i,l-1})^T \in F^l, i = 1, \dots, n$ is the ith coordinate of C. The node repair models can be formalized as follows.

Definition 1 (Centralized model)**.** *Let $\mathcal{F}$ and $\mathcal{R}$ be the sets of failed and helper nodes, and suppose that $|\mathcal{F}| = h \leqslant r$ and $|\mathcal{R}| = d \geqslant k$. We say that the failed nodes $\{C_i, i \in \mathcal{F}\}$ can be repaired from the helper nodes $\{C_j, j \in \mathcal{R}\}$ by downloading*[1] *$\beta_{\mathcal{F}}(\mathcal{R})$ symbols of F if there are d numbers $\beta_j, j \in \mathcal{R}$, d functions $f_j : F^l \to F^{\beta_j}, j \in \mathcal{R}$, and h functions $g_i : F^{\sum_{j\in\mathcal{R}}\beta_j} \to F^l, i \in \mathcal{F}$ such that*

1) for every $i \in \mathcal{F}$ and every $C \in \mathcal{C}$

$$C_i = g_i(\{f_j(C_j), j \in \mathcal{R}\}),$$

2)

$$\sum_{j\in\mathcal{R}} \beta_j = \beta_{\mathcal{F}}(\mathcal{R}).$$

Under the cooperative model, the repair process is divided into two rounds. In the first round, each failed node downloads data from the helper nodes, and in the second round, the failed nodes exchange data among themselves (namely, each failed node downloads data from the other failed nodes).

Definition 2 (Cooperative model)**.** *In the notation of the previous definition, we assume two rounds of communication between the nodes. In the first round, each failed node $C_i, i \in \mathcal{F}$ downloads a vector $f_{ij}(C_j)$ from each helper node $C_j, j \in \mathcal{R}$, and in the second round, each failed node $C_i, i \in \mathcal{F}$ downloads a vector $f_{ii'}(\{f_{i'j}(C_j), j \in \mathcal{R}\})$ from each of the other failed nodes $C_{i'}, i' \in \mathcal{F}\backslash\{i\}$. We require that each failed node $C_i, i \in \mathcal{F}$ can be recovered from its own downloaded data $f_{ij}(C_j), j \in \mathcal{R}$ and $f_{ii'}(\{f_{i'j}(C_j), j \in \mathcal{R}\}), i' \in \mathcal{F}\backslash\{i\}$. The amount of downloaded data in this two-round repair process is*

$$\sum_{i\in\mathcal{F}} \Big(\sum_{j\in\mathcal{R}} \dim_F \big(f_{ij}(C_j)\big) + \sum_{i'\in\mathcal{F}\backslash\{i\}} \dim_F \big(f_{ii'}(\{f_{i'j}(C_j), j \in \mathcal{R}\})\big)\Big),$$

where $\dim_F(\cdot)$ is the dimension of the argument expressed as a vector over F.

This definition may look somewhat restrictive in the part where the communication is constrained to only two rounds. Indeed, in the definition proposed in [16], the repair process may include an arbitrary number T of communication rounds. However, in this paper we show that it suffices to consider $T = 2$ to construct codes with optimal repair bandwidth for all possible parameters, and therefore we rely on the above definition, which also leads to simplified notation. At the same time, it may be that for other problems of cooperative repair, such as optimal-access repair or others, more than two rounds are in fact necessary.

Given a code $\mathcal{C}$, define $N_{\text{ce}}(\mathcal{C}, \mathcal{F}, \mathcal{R})$ and $N_{\text{co}}(\mathcal{C}, \mathcal{F}, \mathcal{R})$ as the smallest number of symbols of F one needs to download in order to recover the failed nodes $\{C_i, i \in \mathcal{F}\}$ from the helper nodes $\{C_j, j \in \mathcal{R}\}$ under the centralized model and the cooperative model, respectively. The repair bandwidth of the code is defined as follows.

Definition 3 (Repair bandwidth)**.** *Let $\mathcal{C}$ be an (n, k, l) MDS array code over a finite field F. The (h, d)-*repair bandwidth *of the code $\mathcal{C}$ under centralized/cooperative repair model is given by*

$$\begin{aligned} \beta_{\text{ce}}(h, d) &:= \max_{|\mathcal{F}|=h,|\mathcal{R}|=d,\mathcal{F}\bigcap\mathcal{R}=\varnothing} N_{\text{ce}}(\mathcal{C}, \mathcal{F}, \mathcal{R}), \\ \beta_{\text{co}}(h, d) &:= \max_{|\mathcal{F}|=h,|\mathcal{R}|=d,\mathcal{F}\bigcap\mathcal{R}=\varnothing} N_{\text{co}}(\mathcal{C}, \mathcal{F}, \mathcal{R}). \end{aligned} \tag{1}$$

As already mentioned, the quantity $\beta(h, d)$ satisfies a general lower bound. In the next theorem we collect results from several papers that establish different versions of this result.

Theorem 1 (Cut-set bound [1], [9], [16], this paper)**.** *Let $\mathcal{C}$ be an (n, k, l) MDS array code. For any two disjoint subsets $\mathcal{F}, \mathcal{R} \subseteq [n]$ such that $|\mathcal{F}| \leqslant r$ and $|\mathcal{R}| \geqslant k$, we have the following inequalities:*

$$N_{\text{ce}}(\mathcal{C}, \mathcal{F}, \mathcal{R}) \geqslant \frac{|\mathcal{F}||\mathcal{R}|l}{|\mathcal{F}| + |\mathcal{R}| - k}, \tag{2}$$

$$N_{\text{co}}(\mathcal{C}, \mathcal{F}, \mathcal{R}) \geqslant \frac{|\mathcal{F}|(|\mathcal{R}| + |\mathcal{F}| - 1)l}{|\mathcal{F}| + |\mathcal{R}| - k}. \tag{3}$$

We note that in [16], the bound (3) was proved under the additional assumption that each failed node downloads the same amount of data from each helper node, and each failed node also downloads the same amount of data from each of the other failed nodes (the *uniform download assumption*), while our proof of (3) in this paper does not require any additional assumptions. A self-contained rigorous proof of (3) is given in Section II as a part of the proof of Theorem 2 below.

Inequality (2) gives the cut-set bound for the centralized model, and (3) gives the cut-set bound under the cooperative one. For the case of a single failed node, there is no difference between the two repair models, and these bounds coincide.

Note that although in this paper we consider only two-round cooperative repair schemes, bound (3) holds for cooperative repair with any number of communication rounds. If $\beta_{\text{ce}}(h, d)$ (resp., $\beta_{\text{co}}(h, d)$) meets the bound (2) (resp., (3)) with equality, i.e.,

$$\beta_{\text{ce}}(h, d) = \frac{hdl}{h + d - k}$$

$$\Big(\text{resp., } \beta_{\text{co}}(h, d) = \frac{h(h + d - 1)l}{h + d - k}\Big),$$

[1] We note the use of the application-inspired term "download" for evaluating the functions f_j and making their values available to the failed nodes. This term is used extensively throughout the paper.

we say that the code $\mathcal{C}$ has the (h,d)-*optimal repair property* under the centralized (resp., cooperative) model.

Let us give a heuristic argument in favor of (3) based on the cut-set bound for repairing single erasure. Let i be one of the indices of the failed nodes. Suppose that all the other failed nodes $C_j, j \in \mathcal{F}\backslash\{i\}$ are functional, and we need to repair C_i. Using either (2) or (3) with $|\mathcal{F}|=1$, we see that C_i needs to download at least $l/(|\mathcal{F}|+|\mathcal{R}|-k)$ field symbols from each of the nodes $C_j, j \in \mathcal{R} \cup \mathcal{F}\backslash\{i\}$. Therefore each failed node $C_i, i \in \mathcal{F}$ needs to download at least $(|\mathcal{F}|+|\mathcal{R}|-1)l/(|\mathcal{F}|+|\mathcal{R}|-k)$ symbols of F in total. Thus, if (3) is achievable with equality, then each failed node can be repaired as though all the other failed nodes were functional and available. We note that this argument is not rigorous because the single-erasure cut-set bound is derived under a one-round repair process while the repair process under the cooperative model is divided into two rounds.

The argument in the previous paragraph also suggests that optimality of a code under cooperative repair implies its optimality under centralized repair. We formalize this idea in the next theorem.

Theorem 2 (Cooperative model is stronger than centralized model)**.** *Let $\mathcal{C}$ be an (n,k,l) MDS array code and let $\mathcal{F},\mathcal{R} \subseteq [n]$ be two disjoint subsets such that $|\mathcal{F}| \leqslant r$ and $|\mathcal{R}| \geqslant k$. If*

$$N_{\text{co}}(\mathcal{C},\mathcal{F},\mathcal{R}) = \frac{|\mathcal{F}|(|\mathcal{R}|+|\mathcal{F}|-1)l}{|\mathcal{F}|+|\mathcal{R}|-k}, \tag{4}$$

then

$$N_{\text{ce}}(\mathcal{C},\mathcal{F},\mathcal{R}) = \frac{|\mathcal{F}||\mathcal{R}|l}{|\mathcal{F}|+|\mathcal{R}|-k}. \tag{5}$$

The statement of the theorem holds for cooperative repair schemes with any number $T \geqslant 2$ of communication rounds.

The statement in Theorem 2 is trivially true under the uniform download assumption and in this form it was stated in [10]. In this paper we prove the theorem in Section II under no additional assumptions. The following arguments provide an intuitive explanation of its claim in the case of $T=2$, and they can be easily extended to any T. As mentioned above, for (4) to hold with equality, each failed node $C_i, i \in \mathcal{F}$ should download $l/(|\mathcal{F}|+|\mathcal{R}|-k)$ symbols of F from each of the nodes $C_j, j \in \mathcal{R} \cup (\mathcal{F}\backslash\{i\})$ in the course of the two-round repair process. Therefore, each failed node $C_i, i \in \mathcal{F}$ downloads only $|\mathcal{R}|l/(|\mathcal{F}|+|\mathcal{R}|-k)$ symbols of F in total from all the helper nodes $\{C_j, j \in \mathcal{R}\}$. Switching to the centralized model, we observe that once these symbols are made available to one failed node, they are automatically available to all the other failed nodes at no cost to the bandwidth, and so (5) follows immediately.

According to Theorem 2, MDS codes with (h,d)-optimal repair property under the cooperative model also have the same property under the centralized model. At the same time, it is not known how to transform optimal centralized-repair codes into cooperative-repair codes. This might be the reason why the latter are more difficult to construct. Indeed, while general (h,d)-optimal repair MDS codes for the centralized model are available in several variations [4], [13], [19], MDS codes with the same property under the cooperative model are known only for some special values of h and d. Specifically, the following results appeared in the literature. Paper [16] constructed optimal MDS codes for cooperative repair for the (trivial) case $d=k$, and [17] presented a family of optimal MDS codes for the repair of two erasures in the regime of low rate $k/n \leqslant 1/2$ (more precisely, [17] constructed (n,k) MDS codes with the $(2,d)$-optimal repair property for any n,k,d such that $2k-3 \leqslant d \leqslant n-2$).

Thus, prior to our work, even the existence problem of cooperative MDS codes with the (h,d)-optimal repair property for general values of h and d (apart from the two special cases mentioned above) was an open question[2].

In the rest of the paper we focus on the cooperative model, and, unless stated otherwise, all the concepts and objects mentioned below such as the repair bandwidth, the cut-set bound, etc., implicitly assume this model.

Our results in this work are as follows:

1) We give a complete solution of repairing multiple erasures for all possible parameters. More precisely, given any n,k,h,d such that $2 \leqslant h \leqslant n-d \leqslant n-k-1$, we present an explicit (n,k) MDS code with the (h,d)-optimal repair property. We limit ourselves to the case of $d \geqslant k+1$ because constructions for $d=k$ were already given in [16].
   The size of the underlying finite field is sn for all constructions, where $s := d+1-k$. At the same time, the sub-packetization l is rather large: for $h=2$ we need to take approximately $l = s^{n(n-1)}$, while for general d and h it is approximately $l = s^{h\binom{n}{h}}$. We do not know whether this is necessary or is merely an artifact of our construction.
2) We prove the cut-set bound (3) for the most general case without the uniform download assumption, and we also show that the any MDS code that affords cooperative optimal repair is also optimally repairable under the centralized model (see Theorem 2).

C. Organization of this paper

In Section II, we prove the general versions of the cut-set bound (3) and Theorem 2 without the uniform download assumption.

In Section III we prove a technical lemma which forms the core of the proposed repair schemes. Various versions of this lemma will be used throughout the paper. Moving to the code constructions, we start with the special case of $h=2$ and $d=k+1$ to illustrate the new ideas behind the proposed code families. These results are presented in Section IV. Namely, in Section IV-A we construct MDS codes $\mathcal{C}_{2,k+1}^{(0)}$ that can optimally repair the first two nodes (or any *given* pair of nodes) from any $d=k+1$ helper nodes. In Section IV-B, we use this code as a building block to construct (n,k) MDS codes $\mathcal{C}_{2,k+1}$ with the $(2, d=k+1)$-optimal repair property.

In Section V, we deal with general values of $d, k+1 \leqslant d \leqslant n-2$. Similarly to the above, in Section V-A we construct a code $\mathcal{C}_{2,d}^{(0)}$ that supports optimal repair of the first two nodes,

[2] In [16], the authors showed that the cut-set bound (3) is achievable under the weaker "functional repair" requirement, which does not assume that the repair scheme recovers the exact content of the failed nodes, as opposed to the more prevalent exact repair requirement considered in this paper.

	Repairing the first h nodes		Repairing any h nodes	
Values of $h = \lvert\mathcal{F}\rvert, d = \lvert\mathcal{R}\rvert$	$\lvert F\rvert$	l	$\lvert F\rvert$	l
Sec. IV: $h = 2, d = k + 1$	$n + 2$	3	$2n$	$3^{\binom{n}{2}}$
Sec. V: $h = 2$, any d	$n + 2(s - 1)$	$s^2 - 1$	sn	$(s^2 - 1)^{\binom{n}{2}}$
Sec. VI: any h, $d = k + 1$	$n + h$	$h + 1$	$2n$	$(h + 1)^{\binom{n}{h}}$
Sec. VII: any h, any d	$n + h(s - 1)$	$(h + d - k)(s - 1)^{h-1}$	sn	$((h + d - k)(s - 1)^{h-1})^{\binom{n}{h}}$

TABLE I: We list the parameters (field size, sub-packetization) of the codes constructed in this paper, where $s := d + 1 - k$. In the first of the two pairs of columns the codes are constructed for optimal repair of the *first h nodes only*, while the second pair gives the parameters of codes that can optimally repair *any h* failed nodes.

and in Section V-B we use it as a building block to construct MDS codes $\mathcal{C}_{2,d}$ with the $(2, d)$-optimal repair property for general values of $d, k + 1 \leqslant d \leqslant n - 2$.

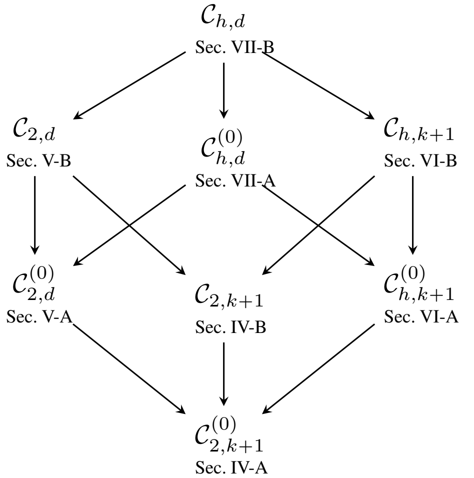

Fig.1: Relations between the code families constructed in the paper. Arrows point from more general code families to their subfamilies. The superscript $^{(0)}$ indicates that the code supports optimal repair of the first two (or the first h) erasures only.

In Section VI we construct (n, k) MDS codes with $(h, d = k + 1)$-optimal repair property for general values of $h, 2 \leqslant h \leqslant r - 1$. Following the route chosen above, in Section VI-A we handle the case of repairing the first h nodes while in Section VI-B we extend the construction to repair any subset of h failed nodes. The corresponding codes are labeled as $\mathcal{C}_{h,k+1}^{(0)}$ and $\mathcal{C}_{h,k+1}$, respectively.

Finally, in Section VII, we present the main result of this paper—the construction for general values of both h and d. In Section VII-A we construct an MDS code $\mathcal{C}_{h,d}^{(0)}$ that supports optimal repair of the first h nodes, and in Section VII-B we use it as a building block to construct an (n, k) MDS codes $\mathcal{C}_{h,d}$ with the (h, d)-optimal repair property for general values of h and d, $2 \leqslant h \leqslant n - d \leqslant r - 1$.

The extension from repairing a fixed h-subset of nodes to any subset of cardinality h relies on an idea that has already appeared in the literature on regenerating codes [4], [19], albeit in a somewhat veiled form. We isolate and illustrate this idea in Section V-C. Apart from revealing the structure behind our constructions, it also enables us to give a family of (n, k) *universal MSR codes* with the (h, d)-optimal repair property for all $1 \leqslant h \leqslant n - d \leqslant n - k$ simultaneously, i.e., these codes can optimally repair any number of failed nodes from any number of helper nodes. This construction forms a simple extension of the main results, and is given in a brief Section VII-C.

Note that Sections IV-VI serve as preparation for Section VII, and all the constructions in Sections IV-VI are special cases of the constructions in Section VII. Even though the structure of the sections looks similar, each of the constructions adds new elements to the basic idea, and without the introductory sections it may be difficult to understand the intuition behind the code constructions in later parts of the paper. At the same time, we note that the codes in Sections VII reduce to the codes in Section V and VI upon appropriate adjustment of the parameters, such as taking $d = k + 1$ or $h = 2$, etc. (see Section VII-A3 below for more details). The complete reduction scheme between the code families in this paper is as shown in Fig. 1, and the parameters of the codes are listed in Table I.

D. Future directions

1) In this paper we consider the problem of repairing multiple erasures for MDS codes, which correspond to the minimum storage regenerating (MSR) point on the trade-off curve between storage and repair bandwidth in the regenerating code literature [1], [20]. A natural future direction is to extend our results to the whole trade-off curve, starting with the minimum bandwidth regenerating (MBR) point.
2) The repair problem of Reed-Solomon (RS) codes has attracted significant attention recently [7], [13], [21]–[27]. In particular, explicit RS code constructions with the (h, d)-optimal repair property under the centralized model were given in [13]. Can this result be extended to the cooperative model (and are two rounds enough)? Note that cooperative repair of (full-length) RS codes was previously considered in [23], which gave schemes for repairing 2 and 3 erasures with small repair bandwidth (since codes in [23] have small l, the repair bandwidth ends up being rather far away from the cut-set bound).
3) Let us consider the regime where we fix the number of parity nodes $r := n - k$ and let n grow. The sub-packetization value of our MDS code construction with the (h, d)-optimal repair property scales as $\exp(\Theta(n^h))$ in this regime, which is much larger than its counterpart under the centralized model, where the sub-packetization value is $\exp(O(n))$ (see [4]). One possible reason is that since the cooperative model is more restrictive than the centralized model, the larger sub-packetization is the penalty we have to pay. The other possibility is that our construction can be improved in terms of the sub-packetization value. This raises an open question of either deriving a lower bound on sub-

packetization for the cooperative model (cf. also Table I) or constructing codes with smaller sub-packetization.

4) Several families of codes under centralized repair also have the *optimal access* property, wherein the number of field symbols accessed at the helper nodes equals the number of symbols downloaded for the purposes of repair [5], [6]. Is it possible to design optimal-repair codes for the cooperative model that reduce or minimize the number of symbols accessed during the repair process?

II. Proof of (3) and Theorem 2

Let $\mathcal{C}$ be an (n,k,l) MDS code over F. Our goal is to prove that if (3) holds with equality, then so does (2). We will argue by showing that inequality (2) implies (3) and then observe that the equality in (3) implies the same for (2). The first step of this argument also yields a self-contained proof of the cooperative cut-set bound (3).

Recall that $h := |\mathcal{F}|$ and $d := |\mathcal{R}|$. To shorten the expressions, below we use the following notation

$$\begin{aligned} D_i(\mathcal{R}) &= \sum_{j\in\mathcal{R}} \dim_F(\big(f_{ij}(C_j)\big), \\ D_i(\mathcal{F}) &= \sum_{i'\in\mathcal{F}\backslash\{i\}} \dim_F\big(f_{ii'}(\{f_{i'j}(C_j), j\in\mathcal{R}\})\big) \end{aligned}$$

for the number of symbols of F downloaded by $C_i \in \mathcal{F}$ from the helper nodes (in the first round of repair) and from the other failed nodes (in the second round of repair), respectively, where the functions $f_{i,\cdot}$ were introduced in Definition 2. For a given node C_i there are $d+h-1$ such functions, and therefore, in total there are $h(d+h-1)$ of them for any given subsets $\mathcal{F}, \mathcal{R}$. Our goal is to show that

$$\sum_{i\in\mathcal{F}} (D_i(\mathcal{R}) + D_i(\mathcal{F})) \geqslant \frac{h(h+d-1)}{h+d-k} l. \tag{6}$$

Our proof relies on the following simple observation: in the first round of the repair process, the data downloaded from the helper nodes by all the failed nodes is the following set of vectors:

$$\{f_{ij}(C_j), i\in\mathcal{F}, j\in\mathcal{R}\}. \tag{7}$$

After obtaining this set of vectors, the failed nodes can recover their values by performing additional information exchange during the second round of repair. Recalling the centralized model, this means that all the information needed to collectively repair the failed nodes is contained in the set (7). Therefore, on account of the centralized version of the cut-set bound (2) we have

$$\sum_{i\in\mathcal{F}} D_i(\mathcal{R}) \geqslant \frac{hd}{h+d-k} l. \tag{8}$$

To bound the second term on the left-hand side of (6), we use the following basic fact about MDS code: for an (n,k) MDS code, any subset of $k-1$ coordinates contains no information about any other coordinate of the code. Assume a uniform distribution on the codewords $C=(C_1,\dots,C_n)\in\mathcal{C}$ and (by a slight abuse of notation) use the same symbols $C_i, i=1,\dots,n$ for the associated random variables. For any $i\in[n]$ (in particular, for any $i\in\mathcal{F}$) and any subset $\mathcal{S}\subseteq\mathcal{R}$ of the helper nodes of size $|\mathcal{S}|=k-1$, we have

$$H(C_i) = H(C_i|\{C_j, j\in\mathcal{S}\}) = l\log_2|F|,$$

where $H(X|Y)$ is the conditional entropy of X given Y, measured in bits. Applying a deterministic function to Y can only increase the conditional entropy, and therefore for any $\mathcal{S}\subseteq\mathcal{R}, |\mathcal{S}|=k-1$ we have

$$H(C_i|\{f_{ij}(C_j), j\in\mathcal{S}\}) = l\log_2(|F|). \tag{9}$$

On the other hand, each $C_i, i\in\mathcal{F}$ is uniquely determined by $\{f_{ij}(C_j), j\in\mathcal{R}\} \cup \{f_{ii'}(\{f_{i'j}(C_j), j\in\mathcal{R}\}) : i'\in\mathcal{F}\backslash\{i\}\}$, so

$$\begin{aligned} H\big(C_i|\{f_{ij}(C_j), j\in\mathcal{R}\} \\ \cup \{f_{ii'}(\{f_{i'j}(C_j), j\in\mathcal{R}\}) : i'\in\mathcal{F}\backslash\{i\}\}\big) = 0. \end{aligned} \tag{10}$$

Combining (9) and (10), and using Lemma 1 below, we obtain that

$$\begin{aligned} H\big(\{f_{ij}(C_j), j\in\mathcal{R}\backslash\mathcal{S}\} \\ \cup \{f_{ii'}(\{f_{i'j}(C_j), j\in\mathcal{R}\}) : i'\in\mathcal{F}\backslash\{i\}\}\big) \\ \geqslant l\log_2|F|. \end{aligned} \tag{11}$$

Therefore, for any $i\in\mathcal{F}$ and any $\mathcal{S}\subseteq\mathcal{R}, |\mathcal{S}|=k-1$

$$\begin{aligned} \sum_{j\in\mathcal{R}\backslash\mathcal{S}} \dim_F\big(f_{ij}(C_j)\big) \\ + \sum_{i'\in\mathcal{F}\backslash\{i\}} \dim_F\big(f_{ii'}(\{f_{i'j}(C_j), j\in\mathcal{R}\})\big) \geqslant l \end{aligned} \tag{12}$$

(the left-hand side on the above line is the entropy of the left-hand side of (11) under the uniform distribution on its arguments. Since the entropy is maximized for the uniform distribution, (12) is implied by (11). Note also the switching of the base of logarithms from 2 to $|F|$.).

Let us sum (12) over all subsets $\mathcal{S}\subseteq\mathcal{R}$ of size $|\mathcal{S}|=k-1$. Only the first term on the left-hand side depends on $\mathcal{S}$, and for every $j\in\mathcal{R}$, the term $\dim_F\big(f_{ij}(C_j)\big)$ appears for $\binom{d-1}{k-1}$ different choices of $\mathcal{S}$. Thus we have

$$\binom{d-1}{k-1} D_i(\mathcal{R}) + \binom{d}{k-1} D_i(\mathcal{F}) \geqslant \binom{d}{k-1} l, \quad i\in\mathcal{F}.$$

Dividing both sides by $\binom{d}{k-1}$, we obtain that for every $i\in\mathcal{F}$,

$$\frac{d-k+1}{d} D_i(\mathcal{R}) + D_i(\mathcal{F}) \geqslant l.$$

Let us sum these inequalities on all $i\in\mathcal{F}$. We obtain

$$\frac{d-k+1}{d} \sum_{i\in\mathcal{F}} D_i(\mathcal{R}) + \sum_{i\in\mathcal{F}} D_i(\mathcal{F}) \geqslant hl. \tag{13}$$

Multiplying (8) on both sides by $\frac{k-1}{d}$ and then adding it to (13), we obtain the desired inequality (6). This completes the proof of (3).

We are left to prove the claim that for a given code $\mathcal{C}$, (4) implies (5). Assuming (4), we observe that there is a choice of the functions $\{\{f_{ij}, j\in\mathcal{R}\}, \{f_{ii'}, i'\in\mathcal{F}\backslash\{i\}\} : i\in\mathcal{F}\}$ such that (6) holds with equality. This means that (13) and all the inequalities preceding it in the proof, including (8), hold with equality, but equality in (8) means that (5) holds true.

Lemma 1. *Let X, Y, Z be arbitrary discrete random variables such that $H(X|YZ) = 0$, then $H(Z) \geqslant H(X|Y)$.*

Proof: By the assumption we have $H(XYZ) = H(YZ)$. Therefore,

$$\begin{aligned} H(Z) \geqslant H(Z|Y) &= H(YZ) - H(Y) \\ &= H(XYZ) - H(Y) \\ &\geqslant H(XY) - H(Y) \\ &= H(X|Y). \end{aligned}$$

■

It remains to justify the final claim of the theorem, namely that it holds for the general case of $T \geqslant 2$ communication rounds. Indeed the proof given above can be easily modified to cover the general situation. To explain this, let us assume that the repair process is divided into T rounds for some finite integer T. In this case, for $i \in \mathcal{F}$ and $j \in \mathcal{R}$, we view $f_{ij}(C_j)$ as all the data downloaded by the failed node C_i from the helper node C_j in all T rounds of communication. For $i, i' \in \mathcal{F}, i \neq i'$, we view $f_{ii'}(\{f_{i'j}(C_j), j \in \mathcal{R}\})$ as all the data downloaded by the failed node C_i from another failed node $C_{i'}$ in all T rounds of communication[3]. It is easy to check that under this point of view, our proof applies directly to a T-round repair process for any integer T.

III. A technical lemma

In this section we prove a technical lemma which will be frequently used throughout the paper. Let $C \in \mathcal{C}$ be a codeword of an $(n, k = n - r, l)$ MDS array code $\mathcal{C}$. We write C as $(C_1, C_2, \dots, C_n)$, where $C_i = (c_{i,0}, c_{i,1}, \dots, c_{i,l-1})^T \in F^l$ is the ith coordinate of C.

Lemma 2. *Let n, k, d be positive integers such that $k \leqslant d \leqslant n - 1$. Let $r := n - k$ and let $s := d + 1 - k$. Let F be a finite field with cardinality $|F| \geqslant n + s - 1$. Let $\lambda_{1,0}, \lambda_{1,1}, \dots, \lambda_{1,s-1}, \lambda_2, \lambda_3, \dots, \lambda_n$ be $n + s - 1$ distinct elements of F. Define an (n, k, s) MDS array code $\mathcal{C}$ over the field F by the following rs parity check equations:*

$$\lambda_{1,u}^t c_{1,u} + \sum_{i=2}^{n} \lambda_i^t c_{i,u} = 0, \quad u = 0, 1, \dots, s-1, \qquad t = 0, 1, \dots, r-1. \tag{14}$$

Let $\mu_i := \sum_{u=0}^{s-1} c_{i,u}$ for all $i \in [n]$. Then for any subset $\mathcal{R} \subseteq \{2, 3, \dots, n\}$ with cardinality $|\mathcal{R}| = d$, the values $\{c_{1,0}, c_{1,1}, \dots, c_{1,s-1}, \mu_2, \mu_3, \dots, \mu_n\}$ can be calculated from $\{\mu_i : i \in \mathcal{R}\}$.

Proof: [4] Summing (14) over $u \in \{0, 1, \dots, s-1\}$, we obtain

$$\sum_{u=0}^{s-1} \lambda_{1,u}^t c_{1,u} + \sum_{i=2}^{n} \lambda_i^t \mu_i = 0, \quad t = 0, 1, \dots, r-1.$$

Writing these r equations in matrix form, we obtain equality (15).

Since $\lambda_{1,0}, \lambda_{1,1}, \dots, \lambda_{1,s-1}, \lambda_2, \lambda_3, \lambda_4, \dots, \lambda_n$ are all distinct, the vector $(c_{1,0}, c_{1,1}, \dots, c_{1,s-1}, \mu_2, \mu_3, \dots, \mu_n)$ is a codeword in an $(n + s - 1, n + s - 1 - r = d)$ generalized Reed-Solomon code. Therefore, for any $\mathcal{R} \subseteq \{2, 3, \dots, n\}, |\mathcal{R}| = d$, the values $\{c_{1,0}, c_{1,1}, \dots, c_{1,s-1}, \mu_2, \mu_3, \dots, \mu_n\}$ can be calculated from $\{\mu_i : i \in \mathcal{R}\}$. This completes the proof of the lemma. ■

IV. Cooperative $(2, k+1)$-optimal codes

A. Repairing the first two nodes from any $k + 1$ helper nodes

Let F be a finite field. For any $k < n \leqslant |F| - 2$ we present a construction of $(n, k, 3)$ MDS array codes $\mathcal{C} = \mathcal{C}_{2,k+1}^{(0)}$ over F that support optimal repair of the first two nodes. Specifically, when the first two nodes of $\mathcal{C}$ fail, the repair of each failed node can be accomplished by connecting to *any* $k + 1$ helper nodes and downloading a total of $k + 2$ symbols of F from these helper nodes as well as from the other failed node, achieving the optimal repair bandwidth according to the cut-set bound (3).

For $i = 1, 2, \dots, n$, we write the ith node of $\mathcal{C}$ as $C_i = (c_{i,0}, c_{i,1}, c_{i,2})^T \in F^3$, which is a column vector of dimension 3 over F. Let $\lambda_{1,0}, \lambda_{1,1}, \lambda_{2,0}, \lambda_{2,1}, \lambda_3, \lambda_4, \dots, \lambda_n$ be $n + 2$ distinct elements of the field F. The code $\mathcal{C}$ is defined by the following 3 sets of parity check equations:

$$\lambda_{1,0}^t c_{1,0} + \lambda_{2,0}^t c_{2,0} + \sum_{i=3}^{n} \lambda_i^t c_{i,0} = 0, \qquad t = 0, 1, \dots, r-1, \tag{16}$$

$$\lambda_{1,1}^t c_{1,1} + \lambda_{2,0}^t c_{2,1} + \sum_{i=3}^{n} \lambda_i^t c_{i,1} = 0, \qquad t = 0, 1, \dots, r-1, \tag{17}$$

$$\lambda_{1,0}^t c_{1,2} + \lambda_{2,1}^t c_{2,2} + \sum_{i=3}^{n} \lambda_i^t c_{i,2} = 0, \qquad t = 0, 1, \dots, r-1. \tag{18}$$

For each $a = 0, 1, 2$ the set of vectors $\{(c_{1,a}, c_{2,a}, \dots, c_{n,a})\}$ obviously forms an $(n, k = n - r)$ MDS code, and so $\mathcal{C}$ is indeed an $(n, k, 3)$ MDS array code.

The following lemma suggests a description of the repair scheme for the first two nodes using the bandwidth that meets the cut-set bound (3) with equality.

Lemma 3. *For $i = 1, \dots, n$ let*

$$\mu_{i,1} := c_{i,0} + c_{i,1}, \quad \mu_{i,2} := c_{i,0} + c_{i,2}.$$

For any set of helper nodes $\mathcal{R} \subseteq \{3, 4, \dots, n\}, |\mathcal{R}| = k + 1$, the values of $c_{1,0}, c_{1,1}$, and $\mu_{2,1}$ are uniquely determined by $\{\mu_{i,1} : i \in \mathcal{R}\}$. Similarly, the values of $c_{2,0}, c_{2,2}$, and $\mu_{1,2}$ are uniquely determined by $\{\mu_{i,2} : i \in \mathcal{R}\}$.

Proof: This lemma follows immediately from Lemma 2. Indeed, take $d = k + 1$ and $s = 2$, then there are only two groups of equations in (14), namely those for $u = 0, 1$. To prove the first statement of Lemma 3, consider the equations in (16) and (17). These two sets of equations have the same structure as the equations in (14): namely, only the coefficients of $c_{1,u}$ vary with u while the coefficients of $c_{i,u}$

[3] Observe that the notation $f_{ii'}(\{f_{i'j}(C_j), j \in \mathcal{R}\})$ is not accurate for multiple-round repair because $f_{ii'}$ can also depend on the data $f_{i'j}, j \in \mathcal{F}\backslash\{i'\}$ downloaded in previous round(s). At the same time, this issue does not affect our argument, so we prefer to keep the already established notation.

[4] This proof draws on the ideas in [4, Theorem 7].

$$\begin{bmatrix} 1 & 1 & \dots & 1 & 1 & 1 & 1 & \dots & 1 \\ \lambda_{1,0} & \lambda_{1,1} & \dots & \lambda_{1,s-1} & \lambda_2 & \lambda_3 & \lambda_4 & \dots & \lambda_n \\ \lambda_{1,0}^2 & \lambda_{1,1}^2 & \dots & \lambda_{1,s-1}^2 & \lambda_2^2 & \lambda_3^2 & \lambda_4^2 & \dots & \lambda_n^2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \lambda_{1,0}^{r-1} & \lambda_{1,1}^{r-1} & \dots & \lambda_{1,s-1}^{r-1} & \lambda_2^{r-1} & \lambda_3^{r-1} & \lambda_4^{r-1} & \dots & \lambda_n^{r-1} \end{bmatrix} \begin{bmatrix} c_{1,0} \\ c_{1,1} \\ \vdots \\ c_{1,s-1} \\ \mu_2 \\ \mu_3 \\ \mu_4 \\ \vdots \\ \mu_n \end{bmatrix} = 0. \tag{15}$$

are independent of the value of u for all $i \in \{2, 3, \dots, n\}$. Therefore Lemma 2 applies directly, and we obtain the claimed fact about $c_{1,0}, c_{1,1}$ and $\mu_{2,1}$.

Similarly, to prove the second statement, consider the equations in (16) and (18). These two sets of equations also have the same structure as the equations in (14): namely, only the coefficients of $c_{2,u}$ vary with u while the coefficients of $c_{i,u}$ are independent of the value of u for all $i \in [n] \backslash \{2\}$. ■

This lemma implies that the first two nodes of $\mathcal{C}$ can be repaired with optimal bandwidth. As already mentioned, the repair process is divided into two rounds. In the first round, the node $C_j, j = 1, 2$ downloads $k+1$ symbols μ_{ij} from the helper nodes $C_i, i \in \mathcal{R}$. According to Lemma 3, after the first round, C_1 knows the values of $c_{1,0}, c_{1,1}$ and $c_{2,0}+c_{2,1}$, and C_2 knows the values of $c_{2,0}, c_{2,2}$ and $c_{1,0} + c_{1,2}$. In the second round, C_1 downloads the sum $c_{1,0}+c_{1,2}$ from C_2, and C_2 downloads the sum $c_{2,0} + c_{2,1}$ from C_1. Clearly, after the second round, both C_1 and C_2 can recover all their coordinates. Moreover, in the whole repair process, C_1 only downloads one symbol of F from each of the nodes $C_i, i \in \mathcal{R} \cup \{2\}$, and C_2 only downloads one symbol of F from each of the nodes $C_i, i \in \mathcal{R} \cup \{1\}$. Therefore the total repair bandwidth is $2(k + 1) + 2$, meeting the cut-set bound (3) with equality.

B. Repairing any two erasures from any $k + 1$ helper nodes

Here we develop the idea in the previous section to construct explicit MDS array codes with the $(2, k + 1)$-optimal repair property. More specifically, given any $n \geqslant k + 3$ and a finite field $F, |F| \geqslant 2n$, we present an $(n, k, l = 3^m)$ MDS array code $\mathcal{C} = \mathcal{C}_{2,k+1}$ over F, where $m = \binom{n}{2}$. When *any* two nodes of $\mathcal{C}$ fail, the repair of each failed node can be accomplished by connecting to *any* $k + 1$ helper nodes and downloading $(k + 2)3^{m-1}$ symbols of F in total from these helper nodes as well as from the other failed node. Clearly, the repair bandwidth meets the cut-set bound (3) with equality.

We will define $\mathcal{C}$ by its parity-check equations, and we begin with some notation. Let $\{\lambda_{i,j}\}_{i\in[n], j\in\{0,1\}}$ be $2n$ distinct elements of the field F. Let g be a bijection between the set of pairs $\{(i_1, i_2) : 1 \leqslant i_1 < i_2 \leqslant n\}$ and the set $\{1, 2, \dots, m\}$. For concreteness, let

$$g : (i_1, i_2) \mapsto \binom{i_2 - 1}{2} + i_1 \tag{19}$$

(g partitions the set $[m]$ into segments of length (i_2-1), where $i_2 = 2, 3, \dots, n$). Given an integer $a \in \{0, 1, \dots, l - 1\}$, let $(a_m, a_{m-1}, \dots, a_1)$ be the digits of its ternary expansion, i.e., $a = \sum_{j=0}^{m-1} a_{j+1} 3^j$. Define the following function

$$\begin{aligned} f : [n] \times \{0, 1, \dots, l - 1\} &\to \{0, 1\} \\ (i, a) &\mapsto \Big(\sum_{j=1}^{i-1} \mathbb{1}\{a_{g(j,i)} = 2\} \\ &\quad + \sum_{j=i+1}^{n} \mathbb{1}\{a_{g(i,j)} = 1\} \Big) \pmod 2, \end{aligned} \tag{20}$$

where $\mathbb{1}$ is the indicator function. We note that f computes the parity of the count of 1's and 2's in a certain subset of the digits of a. This subset is formed of all the digits with indices in the set $\{g(1, i), \dots, g(i - 1, i), g(i, i + 1), \dots, g(i, n)\}$. To give an example, let $n = 6$, then $m = 15$, and the function g maps from $\{(i_1, i_2) : 1 \leqslant i_1 < i_2 \leqslant 6\}$ to $\{1, 2, \dots, 15\}$. Let $i = 2$ and let $0 \leqslant a \leqslant 3^{15} - 1 = 14348906$ be an integer. The function f isolates the digits a_u in the ternary expansions of a such that $u \in \{g(\cdot, 2), g(2, \cdot)\}$, i.e., $u \in \{g(1, 2), g(2, 3), g(2, 4), g(2, 5), g(2, 6)\} = \{1, 3, 5, 8, 12\}$. The value of the function $f(2, a)$ equals the parity of $\mathbb{1}\{a_1 = 2\} + \mathbb{1}\{a_3 = 1\} + \mathbb{1}\{a_5 = 1\} + \mathbb{1}\{a_8 = 1\} + \mathbb{1}\{a_{12} = 1\}$.

Definition 4. *The code $\mathcal{C} = \mathcal{C}_{2,k+1}$ is defined by the following rl parity check equations:*

$$\sum_{i=1}^{n} \lambda_{i,f(i,a)}^t c_{i,a} = 0,$$
$$t = 0, 1, \dots, r - 1, a = 0, 1, \dots, l - 1.$$

For all $a = 0, 1, \dots, l - 1$, the set of vectors $\{(c_{1,a}, c_{2,a}, \dots, c_{n,a})\}$ forms an (n, k) MDS code, so $\mathcal{C}$ is indeed an (n, k, l) MDS array code.

Next we show that $\mathcal{C}$ has optimal repair bandwidth for repairing any two failed nodes from any $k + 1$ helper nodes. Let C_{i_1} and $C_{i_2}, i_1 < i_2$ be the failed nodes. First let us introduce some notation to describe the repair scheme. For $a = 0, 1, \dots, l - 1$, $j \in [m]$, and $u = 0, 1, 2$, let

$$a(j, u) := (a_m, \dots, a_{j+1}, u, a_{j-1}, \dots, a_1).$$

For $a = 0, 1, \dots, l - 1$ and $i \in [n]$, let

$$\mu_{i,1}^{(a)} := c_{i,a(g_{12},0)} + c_{i,a(g_{12},1)},$$
$$\mu_{i,2}^{(a)} := c_{i,a(g_{12},0)} + c_{i,a(g_{12},2)},$$

where for brevity we write g_{12} instead of $g(i_1, i_2)$.

The following lemma, which develops the ideas in Lemma 3, accounts for the $(2, k+1)$ optimal repair property of the code $\mathcal{C}$.

Lemma 4. *Let C_{i_1} and C_{i_2}, $i_1 < i_2$ be the failed nodes. For any set of helper nodes $\mathcal{R} \subseteq [n] \backslash \{i_1, i_2\}, |\mathcal{R}| = k+1$ and any $a \in \{0, 1, \dots, l-1\}$, the values $c_{i_1,a(g_{12},0)}, c_{i_1,a(g_{12},1)}, \mu_{i_2,1}^{(a)}$ are uniquely determined by the set of values $\{\mu_{i,1}^{(a)} : i \in \mathcal{R}\}$. Similarly, the values $c_{i_2,a(g_{12},0)}, c_{i_2,a(g_{12},2)}, \mu_{i_1,2}^{(a)}$ are uniquely determined by the set of values $\{\mu_{i,2}^{(a)} : i \in \mathcal{R}\}$.*

Proof: Recall that $a = 0, 1, \dots, l-1$ numbers the coordinates of the node, or the rows in the codeword array. For a fixed value of a, the parity check equations corresponding to the rows $a(g_{12}, 0), a(g_{12}, 1), a(g_{12}, 2)$ are as follows:

$$\sum_{i=1}^{n} \lambda_{i,f(i,a(g_{12},u))}^{t} c_{i,a(g_{12},u)} = 0, \qquad t = 0, 1, 2, \dots, r-1, u = 0, 1, 2. \tag{21}$$

According to definition of the function f in (20) and the remarks made after it, we have

$$\begin{aligned} f(i, a(g_{12}, 0)) = f(i, a(g_{12}, 1)) &= f(i, a(g_{12}, 2)), \\ & \qquad i \in [n] \backslash \{i_1, i_2\} \\ f(i_1, a(g_{12}, 0)) = f(i_1, a(g_{12}, 2)) &\neq f(i_1, a(g_{12}, 1)), \\ f(i_2, a(g_{12}, 0)) = f(i_2, a(g_{12}, 1)) &\neq f(i_2, a(g_{12}, 2)). \end{aligned}$$

This implies that for $i \in [n] \backslash \{i_1, i_2\}$ the following notation is well defined:

$$\lambda_i := \lambda_{i,f(i,a(g_{12},0))} = \lambda_{i,f(i,a(g_{12},1))} = \lambda_{i,f(i,a(g_{12},2))}. \tag{22}$$

Note that λ_i depends on the value of a, though we omit this dependence from the notation. Further, let

$$\begin{aligned} \lambda'_{i_1,0} &:= \lambda_{i_1,f(i_1,a(g_{12},0))} = \lambda_{i_1,f(i_1,a(g_{12},2))}, \\ \lambda'_{i_1,1} &:= \lambda_{i_1,f(i_1,a(g_{12},1))}, \\ \lambda'_{i_2,0} &:= \lambda_{i_2,f(i_2,a(g_{12},0))} = \lambda_{i_2,f(i_2,a(g_{12},1))}, \\ \lambda'_{i_2,1} &:= \lambda_{i_2,f(i_2,a(g_{12},2))}. \end{aligned} \tag{23}$$

Notice that

$$\begin{gathered} \lambda'_{i_1,0} \neq \lambda'_{i_1,1}, \lambda'_{i_2,0} \neq \lambda'_{i_2,1} \\ \{\lambda'_{i_1,0}, \lambda'_{i_1,1}\} = \{\lambda_{i_1,0}, \lambda_{i_1,1}\} \\ \{\lambda'_{i_2,0}, \lambda'_{i_2,1}\} = \{\lambda_{i_2,0}, \lambda_{i_2,1}\} \\ \lambda_i \in \{\lambda_{i,0}, \lambda_{i,1}\}, \ i \in [n] \backslash \{i_1, i_2\}. \end{gathered}$$

Therefore $\lambda'_{i_1,0}, \lambda'_{i_1,1}, \lambda'_{i_2,0}, \lambda'_{i_2,1}, \lambda_i, i \in [n] \backslash \{i_1, i_2\}$ are all distinct. Using the notation defined in (22)-(23), we can write (21) as

$$\begin{aligned} (\lambda'_{i_1,0})^t c_{i_1,a(g_{12},0)} &+ (\lambda'_{i_2,0})^t c_{i_2,a(g_{12},0)} \\ &+ \sum_{i \in [n] \backslash \{i_1,i_2\}} \lambda_i^t c_{i,a(g_{12},0)} = 0, \\ (\lambda'_{i_1,1})^t c_{i_1,a(g_{12},1)} &+ (\lambda'_{i_2,0})^t c_{i_2,a(g_{12},1)} \\ &+ \sum_{i \in [n] \backslash \{i_1,i_2\}} \lambda_i^t c_{i,a(g_{12},1)} = 0, \\ (\lambda'_{i_1,0})^t c_{i_1,a(g_{12},2)} &+ (\lambda'_{i_2,1})^t c_{i_2,a(g_{12},2)} \\ &+ \sum_{i \in [n] \backslash \{i_1,i_2\}} \lambda_i^t c_{i,a(g_{12},2)} = 0, \\ & \qquad t = 0, 1, 2, \dots, r-1. \end{aligned}$$

Now notice that up to a notational change, these equations have the same form as equations (16)-(18). Therefore, the proof of Lemma 3 applies directly, completing the proof. ■

This lemma implies that the nodes C_{i_1} and C_{i_2} can be repaired with optimal bandwidth. To see this, we partition the coordinates of a node into $l/3$ groups of size 3 where each group is formed of the coordinates with indices $a(g_{12}, 0), a(g_{12}, 1), a(g_{12}, 2)$ for a given a. By Lemma 4 above we know that each group can be repaired with optimal bandwidth, so the entire contents of the failed nodes can also be optimally recovered.

A more detailed description of the repair process is as follows. In the first round of the repair process, C_{i_1} downloads the values in the set $\{\mu_{i,1}^{(a)} : a_{g_{12}} = 0\}$ and C_{i_2} downloads the values $\{\mu_{i,2}^{(a)} : a_{g_{12}} = 0\}$ from each helper node $C_i, i \in \mathcal{R}$. This enables C_{i_1} to find the values

$$\{c_{i_1,a} : a_{g_{12}} = 0\} \cup \{c_{i_1,a(g_{12},1)} : a_{g_{12}} = 0\} \cup \{\mu_{i_2,1}^{(a)} : a_{g_{12}} = 0\}.$$

Similarly, C_{i_2} is able to find the values

$$\{c_{i_2,a} : a_{g_{12}} = 0\} \cup \{c_{i_2,a(g_{12},2)} : a_{g_{12}} = 0\} \cup \{\mu_{i_1,2}^{(a)} : a_{g_{12}} = 0\}.$$

In the second round, C_{i_1} downloads $\{\mu_{i_1,2}^{(a)} : a_{g_{12}} = 0\}$ from C_{i_2}, and C_{i_2} downloads $\{\mu_{i_2,1}^{(a)} : a_{g_{12}} = 0\}$ from C_{i_1}. After the second round, C_{i_1} knows the values of all the elements in the set

$$\begin{aligned} &\{c_{i_1,a(g_{12},u)} : a_{g_{12}} = 0, u \in \{0, 1, 2\}\} \\ &\quad = \{c_{i_1,a} : a \in \{0, 1, 2, \dots, l-1\}\}, \end{aligned}$$

and C_{i_2} knows the values of all the elements in the set

$$\begin{aligned} &\{c_{i_2,a(g_{12},u)} : a_{g_{12}} = 0, u \in \{0, 1, 2\}\} \\ &\quad = \{c_{i_2,a} : a \in \{0, 1, 2, \dots, l-1\}\}, \end{aligned}$$

i.e., both C_{i_1} and C_{i_2} can recover all their coordinates. Moreover, in the whole repair process, C_{i_1} downloads $l/3$ symbols of F from each of the nodes $C_i, i \in \mathcal{R} \cup \{i_2\}$, and C_{i_2} downloads $l/3$ symbols of F from each of the nodes $C_i, i \in \mathcal{R} \cup \{i_1\}$. Therefore the total repair bandwidth is $2(k+2)l/3$, meeting the cut-set bound (3) with equality.

V. Cooperative $(2, d)$-optimal codes for general d

A. Optimal repair of the first two nodes

In this section we present an explicit MDS array code that can optimally repair the first two nodes from any d helper nodes for general values of d. Let n, k, d be such that $k+1 \leqslant d \leqslant n-2$, let $s := d+1-k$, and let F be a finite field of size at least $n - 2 + 2s$. We will construct an $(n, k, s^2 - 1)$ MDS array code $\mathcal{C} = \mathcal{C}_{2,d}^{(0)}$ over the field F that has the following property. When the first two nodes of $\mathcal{C}$ fail, the repair of each of them can be accomplished by connecting to *any* d surviving (helper) nodes and downloading $(s-1)(d+1)$ symbols of F in total from these helper nodes as well as from the other failed node. Clearly, the amount of downloaded data meets the cut-set bound (3) with equality.

Let $\lambda_{1,0}, \lambda_{1,1}, \dots, \lambda_{1,s-1}, \lambda_{2,0}, \lambda_{2,1}, \dots, \lambda_{2,s-1}, \lambda_3, \lambda_4, \dots, \lambda_n$ be $n-2+2s$ distinct elements of the field F. Given an integer $a, 0 \leqslant a \leqslant s^2-2$, let $b_1(a), b_2(a)$ be the digits of its expansion to the base s:

$$a = (b_2(a), b_1(a)). \tag{24}$$

The code $\mathcal{C} = \mathcal{C}_{2,d}^{(0)}$ is defined by the following $r(s^2-1)$ parity check equations.

$$\lambda_{1,b_1(a)}^t c_{1,a} + \lambda_{2,b_2(a)}^t c_{2,a} + \sum_{i=3}^n \lambda_i^t c_{i,a} = 0. \tag{25}$$
$$t = 0,1,\dots,r-1,\ a = 0,1,2,\dots,s^2-2.$$

Clearly, for a given a the set of vectors $\{(c_{1,a}, c_{2,a}, \dots, c_{n,a})\}$ that satisfy the system (25) forms an MDS code of length n and dimension k. Therefore $\mathcal{C}$ is indeed an (n,k,s^2-1) MDS array code. Note that for $d = k+1$, the code $\mathcal{C}$ defined by (25) is the same as the code defined by (16)-(18) in Section IV.

For every $i \in [n]$ define the following elements of F:

$$\mu_{i,1}^{(v_2)} := \sum_{v_1=0}^{s-1} c_{i,sv_2+v_1}, \quad v_2 \in \{0,1,\dots,s-2\};$$
$$\mu_{i,2}^{(v_1)} := \sum_{v_2=0}^{s-1} c_{i,sv_2+v_1}, \quad v_1 \in \{0,1,\dots,s-2\}.$$

Similarly to the previous sections, we have the following lemma:

Lemma 5. *Suppose that the failed nodes are C_1, C_2 and let $\mathcal{R} \subseteq \{3,4,\dots,n\}, |\mathcal{R}| = d$ be a set of d helper nodes. For any $v_2 \in \{0,1,\dots,s-2\}$, the values $\{c_{1,sv_2+v_1}, v_1 = 0,1,\dots,s-1\}$ and $\mu_{2,1}^{(v_2)}$ are uniquely determined by the set of values $\{\mu_{i,1}^{(v_2)} : i \in \mathcal{R}\}$. Similarly, for any $v_1 \in \{0,1,\dots,s-2\}$, the values $\{c_{2,sv_2+v_1}, v_2 = 0,1,\dots,s-1\}$ and $\mu_{1,2}^{(v_1)}$ are uniquely determined by the set of values $\{\mu_{i,2}^{(v_1)} : i \in \mathcal{R}\}$.*

Proof: We again use Lemma 2 to prove this lemma. To prove the first statement, we use definition (25) to write out the parity-check equations that correspond to $a = sv_2, sv_2+1, \dots, sv_2+s-1$ for a fixed $v_2 \in \{0,1,\dots,s-2\}$:

$$\lambda_{1,v_1}^t c_{1,sv_2+v_1} + \lambda_{2,v_2}^t c_{2,sv_2+v_1} + \sum_{i=3}^n \lambda_i^t c_{i,sv_2+v_1} = 0,$$
$$t = 0,1,\dots,r-1,\ v_1 = 0,1,\dots,s-1.$$

These equations have the same structure as the equations in (14): v_1 here plays the role of u in (14). Only the coefficients of c_{1,sv_2+v_1} vary with the value of v_1 while the coefficients of c_{i,sv_2+v_1} are independent of the value of v_1 for all $i \in [n]\backslash\{1\}$. Therefore the proof of Lemma 2 can be directly applied here.

To prove the second statement, we use definition (25) to write out the parity-check equations that correspond to $a = v_1, v_2+v_1, 2v_2+v_1, \dots, (s-1)v_2+v_1$ for a fixed $v_1 \in \{0,1,\dots,s-2\}$:

$$\lambda_{1,v_1}^t c_{1,sv_2+v_1} + \lambda_{2,v_2}^t c_{2,sv_2+v_1} + \sum_{i=3}^n \lambda_i^t c_{i,sv_2+v_1} = 0,$$
$$t = 0,1,\dots,r-1,\ v_2 = 0,1,\dots,s-1.$$

These equations have the same structure as the equations in (14): v_2 here plays the role of u in (14). Only the coefficients of c_{2,sv_2+v_1} vary with the value of v_2 while the coefficients of c_{i,sv_2+v_1} are independent of the value of v_2 for all $i \in [n]\backslash\{2\}$. Therefore the proof of Lemma 2 can be directly applied here. ■

Let us show that this lemma implies that the first two nodes of $\mathcal{C}$ can be repaired with optimal bandwidth. In the first round, the first node C_1 downloads the values $\{\mu_{i,1}^{(v_2)}, v_2 = 0,1,\dots,s-2\}$ from each helper node $C_i, i \in \mathcal{R}$, and the second node C_2 downloads $\{\mu_{i,2}^{(v_1)}, v_1 = 0,1,\dots,s-2\}$ from each helper node $C_i, i \in \mathcal{R}$. From Lemma 5 we conclude that after the first round, C_1 knows the values

$$c_{1,sv_2+v_1},\ v_2 = 0,1,\dots,s-2, v_1 = 0,1,\dots,s-1$$
$$\text{and } \mu_{2,1}^{(v_2)},\ v_2 = 0,1,\dots,s-2.$$

In the same way, C_2 knows the values

$$c_{2,sv_2+v_1},\ v_1 = 0,1,\dots,s-2, v_2 = 0,1,\dots,s-1$$
$$\text{and } \mu_{1,2}^{(v_1)},\ v_1 = 0,1,\dots,s-2.$$

In the second round, C_1 downloads the sums $\mu_{1,2}^{(v_1)}, v_1 = 0,1,\dots,s-2$ from C_2, and C_2 downloads the sums $\mu_{2,1}^{(v_2)}, v_2 = 0,1,\dots,s-2$ from C_1. It is easy to verify that after the second round, both C_1 and C_2 can recover all of their coordinates. Moreover, over the course of the entire repair process, C_1 downloads $(s-1)$ symbols of F from each of the nodes $C_i, i \in \mathcal{R} \cup \{2\}$, and C_2 downloads $(s-1)$ symbols of F from each of the nodes $C_i, i \in \mathcal{R} \cup \{1\}$. Therefore the total repair bandwidth is $2(s-1)(d+1)$, meeting the cut-set bound (3) with equality.

B. Optimal repair of any two erasures

In this section we present a construction of MDS array codes with the $(2,d)$-optimal repair property, relying on the ideas of the previous section. Let n, k, d be such that $k+1 \leqslant d \leqslant n-2$, let $s := d+1-k$ and let F be a finite field such that $|F| \geqslant sn$. We present an $(n, k, l = (s^2-1)^m)$ MDS array code $\mathcal{C} = \mathcal{C}_{2,d}$ over the field F, where $m := \binom{n}{2}$. When *any* two nodes of $\mathcal{C}$ fail, the repair of each failed node can be accomplished by connecting to *any* d helper nodes and downloading $(d+1)l/(s+1)$ symbols of F in total from these helper nodes as well as from the other failed node. Clearly, the repair bandwidth meets the cut-set bound (3) with equality.

We will define $\mathcal{C}$ by its parity-check equations, and we begin with some notation. Let $\{\lambda_{ij}\}_{i \in [n], j \in \{0,1,\dots,s-1\}}$ be sn distinct elements of the field F. Let g be a bijection between the set of pairs $\{(i_1,i_2) : i_1, i_2 \in [n], i_1 < i_2\}$ and the set $\{1,2,\dots,m\}$ defined in (19). For every $a = 0,1,2,\dots,l-1$, we write its expansion in the base (s^2-1) as $a = (a_m, a_{m-1}, \dots, a_1)$, i.e., $a = \sum_{j=0}^{m-1} a_{j+1}(s^2-1)^j$. Define the following function

$$f : [n] \times \{0,1,\dots,l-1\} \to \{0,1,\dots,s-1\}$$
$$(i,a) \mapsto \Big(\sum_{j=1}^{i-1} b_2(a_{g(j,i)}) + \sum_{j=i+1}^{n} b_1(a_{g(i,j)})\Big) \pmod{s}, \tag{26}$$

where $b_1(x)$ and $b_2(x)$ form the digits of the expansion of x in the base s; see definition (24). Note that when $d = k+1$, the function f defined in (26) is the same as the function defined in (20) in Section IV-B.

Definition 5. *The code $\mathcal{C} = \mathcal{C}_{2,d}$ is defined by the following rl parity check equations.*

$$\sum_{i=1}^{n} \lambda_{i,f(i,a)}^{t} c_{i,a} = 0, \quad t = 0,1,2,\dots,r-1,$$
$$a = 0,1,2,\dots,l-1.$$

For a given $a = 0,1,\dots,l-1$ the set of vectors $\{(c_{1,a}, c_{2,a}, \dots, c_{n,a})\}$ forms an MDS code of length n and dimension k. Therefore $\mathcal{C}$ is indeed an (n,k,l) MDS array code. Also note that when $d = k+1$, the code $\mathcal{C}$ is the same as the code defined in Section IV-B.

Next we show that $\mathcal{C}$ has optimal repair bandwidth for repairing any two failed nodes from any d helper nodes. We need several elements of notation which are similar to the notation used in the previous sections. For $a = 0,1,\dots,l-1$, $j \in [m]$, and $u \in \{0,1,2,\dots,s^2-2\}$, let $a(j,u) := (a_m, \dots, a_{j+1}, u, a_{j-1}, \dots, a_1)$. For $a = 0,1,\dots,l-1$ and $i \in [n]$, we define

$$\mu_{i,i_1}^{(a,v_2)} := \sum_{v_1=0}^{s-1} c_{i,a(g_{12}, sv_2+v_1)}, \ v_2 = 0,1,\dots,s-2,$$
$$\mu_{i,i_2}^{(a,v_1)} := \sum_{v_2=0}^{s-1} c_{i,a(g_{12}, sv_2+v_1)}, \ v_1 = 0,1,\dots,s-2,$$

where for brevity we again write g_{12} instead of $g(i_1,i_2)$. The following lemma implies that $\mathcal{C}$ is an MDS code with the $(2,d)$ optimal repair property.

Lemma 6. *Let the failed nodes be C_{i_1} and C_{i_2}, $1 \leqslant i_1 < i_2 \leqslant n$ and let $\mathcal{R} \subset [n], |\mathcal{R}| = d$ be a set of d helper nodes. For any $a \in \{0,1,\dots,l-1\}$ and any $v_2 \in \{0,1,\dots,s-2\}$, the values $\{c_{i_1,a(g_{12},sv_2+v_1)}, v_1 = 0,1,\dots,s-1\}$ and $\mu_{i_2,i_1}^{(a,v_2)}$ are uniquely determined by the set of values $\{\mu_{i,i_1}^{(a,v_2)} : i \in \mathcal{R}\}$. Similarly, for any $v_1 \in \{0,1,\dots,s-2\}$, the values $\{c_{i_2,a(g_{12},sv_2+v_1)}, v_2 = 0,1,\dots,s-1\}$ and $\mu_{i_1,i_2}^{(a,v_1)}$ are uniquely determined by the set of values $\{\mu_{i,i_2}^{(a,v_1)} : i \in \mathcal{R}\}$.*

Proof: The parity-check equations that correspond to the row indices $a(g_{12},0), a(g_{12},1), \dots, a(g_{12},s^2-2)$ are as follows:

$$\sum_{i=1}^{n} \lambda_{i,f(i,a(g_{12},u))}^{t} c_{i,a(g_{12},u)} = 0, \quad t = 0,1,2,\dots,r-1, u = 0,1,\dots,s^2-2. \tag{27}$$

According to definition of the function f in (26), if $i \neq i_1, i_2$ then the value of f does not depend on the value of the digit $a_{g_{12}}$. Thus, we have

$$f(i,a(g_{12},0)) = f(i,a(g_{12},1)) = \dots = f(i,a(g_{12},s^2-2)), \quad i \in [n]\backslash\{i_1,i_2\}.$$

Again according to (26), for all $u = 0,1,2,\dots,s^2-2$, we have

$$\begin{aligned} f(i_1,a(g_{12},u)) &= \big(f(i_1,a(g_{12},0)) + b_1(u)\big) \ (\operatorname{mod} s), \\ f(i_2,a(g_{12},u)) &= \big(f(i_2,a(g_{12},0)) + b_2(u)\big) \ (\operatorname{mod} s). \end{aligned} \tag{28}$$

Therefore, we are justified in using the following notation:

$$\begin{aligned} \lambda_i &:= \lambda_{i,f(i,a(g(i_1,i_2),0))} = \lambda_{i,f(i,a(g(i_1,i_2),1))} \\ &\quad = \lambda_{i,f(i,a(g(i_1,i_2),2))}, \ i \notin \{i_1,i_2\}, \end{aligned} \tag{29}$$
$$\lambda'_{i_1,v} := \lambda_{i_1, v \oplus f(i_1,a(g_{12},0))},$$
$$\lambda'_{i_2,v} := \lambda_{i_2, v \oplus f(i_2,a(g_{12},0))}, \ v \in \{0,1,\dots,s-1\},$$

where $\oplus$ is addition modulo s. By (28), for every $u = 0,1,2,\dots,s^2-2$, we have

$$\begin{aligned} \lambda_{i_1,f(i_1,a(g_{12},u))} &= \lambda_{i_1,b_1(u) \oplus f(i_1,a(g_{12},0))} = \lambda'_{i_1,b_1(u)}; \\ \lambda_{i_2,f(i_2,a(g_{12},u))} &= \lambda_{i_2,b_2(u) \oplus f(i_2,a(g_{12},0))} = \lambda'_{i_2,b_2(u)}. \end{aligned} \tag{30}$$

Notice that

$$\{\lambda'_{i,0}, \lambda'_{i,1}, \dots, \lambda'_{i,s-1}\} = \{\lambda_{i,0}, \lambda_{i,1}, \dots, \lambda_{i,s-1}\} \quad \text{for } i \in \{i_1,i_2\},$$

and that

$$\lambda_i \in \{\lambda_{i,0}, \lambda_{i,1}, \dots, \lambda_{i,s-1}\} \text{ for all } i \in [n]\backslash\{i_1,i_2\}.$$

Therefore $\lambda'_{i_1,0}, \lambda'_{i_1,1}, \dots, \lambda'_{i_1,s-1}, \lambda'_{i_2,0}, \lambda'_{i_2,1}, \dots, \lambda'_{i_2,s-1}, \lambda_i$, $i \in [n]\backslash\{i_1,i_2\}$ are all distinct. Using (29) and (30), we can write (27) as

$$(\lambda'_{i_1,b_1(u)})^t c_{i_1,a(g_{12},u)} + (\lambda'_{i_2,b_2(u)})^t c_{i_2,a(g_{12},u)} + \sum_{i \in [n]\backslash\{i_1,i_2\}} \lambda_i^t c_{i,a(g_{12},u)} = 0,$$
$$t = 0,1,2,\dots,r-1, \ u = 0,1,\dots,s^2-2.$$

These equations have exactly the same form as the equations in (25). Therefore the remainder of the proof of this lemma follows the steps in the proof of Lemma 5, and there is no need to reproduce them here. ■

This lemma enables us to set up a repair procedure for the nodes C_{i_1} and C_{i_2}. In the first round of repair, C_{i_1} downloads the set of elements

$$\bigcup_{v_2=0}^{s-2} \{\mu_{i,i_1}^{(a,v_2)} : a_{g_{12}} = 0\} \tag{31}$$

from each helper node $C_i, i \in \mathcal{R}$. In the same way, C_{i_2} downloads the set of elements

$$\bigcup_{v_1=0}^{s-2} \{\mu_{i,i_2}^{(a,v_1)} : a_{g_{12}} = 0\}$$

from each helper node $C_i, i \in \mathcal{R}$. For future use, let us calculate the number of symbols that C_{i_1} downloads from $C_i, i \in \mathcal{R}$, i.e., the cardinality of the set in (31). Since each digit of a in its (s^2-1)-ary expansion can take s^2-1 possible values, $|\{\mu_{i,i_1}^{(a,v_2)} : a_{g_{12}} = 0\}| = l/(s^2-1)$. The set in (31) is the union of $s-1$ such sets, so its cardinality is $(s-1)l/(s^2-1) = l/(s+1)$.

According to Lemma 6, after the first round, C_{i_1} knows the values of

$$\begin{aligned}\Big(\bigcup_{v_2=0}^{s-2}\bigcup_{v_1=0}^{s-1}\{c_{i_1,a(g_{12},sv_2+v_1)}:a_{g_{12}}=0\}\Big)&\\ \bigcup\Big(\bigcup_{v_2=0}^{s-2}\{\mu_{i_2,i_1}^{(a,v_2)}:a_{g_{12}}=0\}\Big)&,\end{aligned}\tag{32}$$

and C_{i_2} knows the values of

$$\begin{aligned}\Big(\bigcup_{v_1=0}^{s-2}\bigcup_{v_2=0}^{s-1}\{c_{i_2,a(g_{12},sv_2+v_1)}:a_{g_{12}}=0\}\Big)&\\ \bigcup\Big(\bigcup_{v_1=0}^{s-2}\{\mu_{i_1,i_2}^{(a,v_1)}:a_{g_{12}}=0\}\Big)&.\end{aligned}\tag{33}$$

In the second round of the repair process, the nodes C_{i_1}, C_{i_2} exchange the second terms in (32)-(33): namely, C_{i_1} downloads the elements in the set $\cup_{v_1=0}^{s-2}\{\mu_{i_1,i_2}^{(a,v_1)} : a_{g_{12}} = 0\}$ from C_{i_2}, and C_{i_2} downloads the elements in the set $\cup_{v_2=0}^{s-2}\{\mu_{i_2,i_1}^{(a,v_2)} : a_{g_{12}} = 0\}$ from C_{i_1}. After the second round, C_{i_1} knows the values of all the elements in the set

$$\begin{aligned}&\{c_{i_1,a(g_{12},u)}:a_{g_{12}}=0,u\in\{0,1,2,\dots,s^2-2\}\}\\ =&\{c_{i_1,a}:a\in\{0,1,2,\dots,l-1\}\},\end{aligned}$$

and C_{i_2} knows the values of all the elements in the set

$$\begin{aligned}&\{c_{i_2,a(g_{12},u)}:a_{g_{12}}=0,u\in\{0,1,2,\dots,s^2-2\}\}\\ =&\{c_{i_2,a}:a\in\{0,1,2,\dots,l-1\}\},\end{aligned}$$

i.e., both C_{i_1} and C_{i_2} have recovered all their coordinates. Moreover, in the course of the repair process, C_{i_1} downloads $l/(s+1)$ symbols of F from each of the nodes $C_i, i \in \mathcal{R}\cup\{i_2\}$, and C_{i_2} downloads $l/(s+1)$ symbols of F from each of the nodes $C_i, i \in \mathcal{R}\cup\{i_1\}$. Therefore the total repair bandwidth is $2(d+1)l/(s+1)$, meeting the cut-set bound (3) with equality.

C. Optimal repair of two erasures from arbitrary number of helper nodes

In this section, we point out a technique which has been used extensively but somewhat implicitly in the literature, and we use it to construct (n, k) MDS array codes with the universal $(2, d)$-optimal repair property for all $k \leqslant d \leqslant n-2$ simultaneously. We only aim to convey the main ideas underlying the universal constructions, and we will not discuss all the details in a rigorous way which would require developing new notation, and would lead to tedious and redundant presentation. The initial idea to use the expansion of the row index is due to [3], [28], and it was used in [4] to construct explicit universal families of regenerating codes for centralized repair.

To illustrate this technique, let us start from the simplest case of repairing single erasure. Returning to the $(n, k, s = d+1-k)$ MDS code defined by the parity-check equations in (14), we observe that the proof of Lemma 3 gives a repair scheme of the first node relying on downloading a $\frac{1}{s}$ proportion of symbols from each of the d helper nodes (it also gives the μ_i's which at this point we ignore). Moreover, as already remarked, with straightforward changes to the construction we can obtain a code with optimal repair of the ith node for any given $i = 1, \dots, n$. Denote this code by $\mathcal{C}_i$.

The next step is to show how two codes of this kind can be combined to construct an $(n, k, l = s^2)$ MDS code that supports optimal repair of each of the first two nodes from any d helper nodes. For instance, take the codes $\mathcal{C}_1, \mathcal{C}_2$ defined over a field F of size at least $n+2s-2$, and let $\lambda_{1,0}, \lambda_{1,1}, \dots, \lambda_{1,s-1}, \lambda_{2,0}, \lambda_{2,1}, \dots, \lambda_{2,s-1}, \lambda_3, \lambda_4, \dots, \lambda_n$ be distinct elements of F. Define an (n, k, s^2) MDS array code $\mathcal{C} = \mathcal{C}_1 \odot \mathcal{C}_2$ over F by the following rs^2 parity-check equations:

$$\begin{gathered}\lambda_{1,a_1}^t c_{1,a}+\lambda_{2,a_2}^t c_{2,a}+\sum_{i=3}^n \lambda_i^t c_{i,a}=0,\\ a=0,1,\dots,s^2-1,\quad t=0,1,\dots,r-1,\end{gathered}\tag{34}$$

where (a_1, a_2) is the two-digit s-ary expansion of the row index $a \in \{0, 1, \dots, s^2-1\}$. For the repair of the first node, we fix a_2 and let a_1 take all the values in the set $\{0, 1, \dots, s-1\}$. In this way we divide the coordinates of each node into s groups according to the value of a_2, and the parity check equations that correspond to each group have exactly the same structure as (14). Therefore we can optimally repair the first node from any d helper nodes. At the same time, fixing a_1 and varying a_2, we can optimally repair the second node in the same way.

It is clear that the code $\mathcal{C}$ defined by (34) is obtained by a combination of the codes $\mathcal{C}_1$ and $\mathcal{C}_2$ which is similar to the so-called serial concatenation [29]. Now it is easily seen that the code $\mathcal{C}_{1,d} := \mathcal{C}_1 \odot \mathcal{C}_2 \odot \dots \odot \mathcal{C}_n$ has the $(1, d)$-optimal repair property. In fact, this code family already appeared in the literature; see Construction 2 in [4].

Now let us consider cooperative repair of two erasures. For $\mathcal{F} \subseteq [n], |\mathcal{F}| = 2$ and $k \leqslant d \leqslant n-2$, let $\mathcal{C}_{\mathcal{F},d}$ be the $(n, k, l = s^2-1)$ MDS array code that can optimally repair the failed nodes $C_i, i \in \mathcal{F}$ from any d helper nodes. Note that $\mathcal{C}_{\{1,2\},d}$ is the code defined by (25), and we previously denoted it as $\mathcal{C}_{2,d}^{(0)}$. As before, the specific choice of $\mathcal{F}$ is not important, and we can construct a code $\mathcal{C}_{\mathcal{F},d}$ with the same structure and parameters as $\mathcal{C}_{\{1,2\},d}$ for any 2-subset $\mathcal{F} \subset [n]$. Now it is clear that the code $\mathcal{C}_{2,d}$ in Definition 5 is the concatenation of all $\mathcal{C}_{\mathcal{F},d}$ such that $\mathcal{F} \subseteq [n], |\mathcal{F}| = 2$, i.e.,

$$\mathcal{C}_{2,d}=\bigodot_{\mathcal{F}\subseteq[n],|\mathcal{F}|=2}\mathcal{C}_{\mathcal{F},d}.$$

Following this line of thought, we can easily construct an (n, k) MDS array code $\mathcal{C}_2^U$ with the *universal* $(2, d)$*-optimal repair property* for all $k \leqslant d \leqslant n-2$ simultaneously. Namely, the concatenated code[5]

$$\mathcal{C}_2^U:=\bigodot_{k+1\leqslant d\leqslant n-2}\mathcal{C}_{2,d}$$

can optimally repair any two failed nodes from any subset of d helper nodes as long as $d \geqslant k$. The size of the finite field is determined by the code $\mathcal{C}_{2,n-2}$ and is at least $(r-1)n$, and the sub-packetization of the code $\mathcal{C}_2^U$ equals $\prod_{d=k+1}^{n-2}\big((d-k+1)^2-1\big)^{\binom{n}{2}}$.

[5] It is easy to see that the code $\mathcal{C}_{2,n-2}$ has the $(2, d)$-optimal repair property not only for $d = n-2$, but also for $d = k$. Therefore in the concatenation we do not need to include $\mathcal{C}_{2,k}$.

VI. Cooperative $(h, k+1)$ optimal codes for general h

A. Repairing the first h nodes from any $d = k+1$ helper nodes

In this section we present a construction of MDS array codes that can optimally repair the first h nodes from any $d = k+1$ helper nodes for any given $h = 2, \dots, r-1$. More specifically, given any $k < n$, any $h \leqslant r-1$, and a finite field F of cardinality $|F| \geqslant n+h$, we present an $(n, k, h+1)$ MDS array code $\mathcal{C} = \mathcal{C}_{h,k+1}^{(0)}$ over the field F that has the following property. When the first h nodes of $\mathcal{C}$ fail, the repair of each failed node can be accomplished by connecting to *any* $k+1$ helper nodes and downloading $k+h$ symbols of F in total from these helper nodes as well as from other failed nodes. Clearly, the amount of downloaded data meets the cut-set bound (3) with equality.

Let $(\lambda_{ij}, i = 1, \dots, h, j = 0, 1), \lambda_{h+1}, \lambda_{h+2}, \dots, \lambda_n$ be $n+h$ distinct elements of the field F. The code $\mathcal{C}$ is defined by the following parity check equations.

$$\begin{gathered}\sum_{i=1}^{h} \lambda_{i,0}^t c_{i,0} + \sum_{i=h+1}^{n} \lambda_i^t c_{i,0} = 0, \quad t = 0, 1, \dots, r-1;\\ \lambda_{a,1}^t c_{a,a} + \sum_{i \in [h] \backslash \{a\}} \lambda_{i,0}^t c_{i,a} + \sum_{i=h+1}^{n} \lambda_i^t c_{i,a} = 0,\\ t = 0, 1, \dots, r-1, \; a = 1, 2, \dots, h.\end{gathered} \tag{35}$$

For every $a = 0, 1, \dots, h,$ the set of vectors $\{(c_{1,a}, c_{2,a}, \dots, c_{n,a})\}$ forms an (n, k) MDS code, therefore $\mathcal{C}$ is indeed an $(n, k, h+1)$ MDS array code. When $h = 2$, this code is the same as the code defined in Section IV.

For $i \in [n]$ and $j \in [h]$, define

$$\mu_{ij} := c_{i,0} + c_{ij}.$$

Similarly to the previous sections, we have the following lemma:

Lemma 7. *Let $C_1, \dots, C_h$ be the failed nodes. For any set of helper nodes $\mathcal{R} \subseteq \{h+1, h+2, \dots, n\}, |\mathcal{R}| = k+1$ and any $j \in [h]$, the values of $c_{j,0}, c_{j,j}$ and the sums $\{\mu_{ij}, i \in [h] \backslash \{j\}\}$ are uniquely determined by $\{\mu_{ij} : i \in \mathcal{R}\}$.*

The proof of this lemma is the same as that of Lemma 3, and we do not repeat it here. This lemma implies that the first h nodes of $\mathcal{C}$ can be repaired with optimal bandwidth. In the first round, every failed node $C_j, j \in [h]$ downloads μ_{ij} from each helper node $C_i, i \in \mathcal{R}$. According to Lemma 7, after the first round, for every $j \in [h]$, the node C_j knows the values of $c_{j,0}, c_{j,j}$ and $\{\mu_{ij}, i \in [h] \backslash \{j\}\}$. In the second round, every failed node $C_j, j \in [h]$ downloads the sum μ_{ji} from each of the other failed nodes $C_i, i \in [h] \backslash \{j\}$. After the second round, every failed node $C_j, j \in [h]$ knows the values of $c_{j,0}, c_{j,j}$ and the sums $c_{j,0} + c_{j,i}, i \in [h] \backslash \{j\}$. Therefore C_j can recover all its coordinates. Moreover, in the whole repair process, every failed node $C_j, j \in [h]$ downloads only one symbol of F from each of the nodes $C_i, i \in \mathcal{R} \cup [h] \backslash \{j\}$. Therefore the total repair bandwidth is $h(k+h)$, meeting the cut-set bound (3) with equality.

B. Repairing arbitrary h nodes

In this section we construct explicit MDS array codes that support $(h, k+1)$-optimal repair of any h-tuple of failed nodes. More specifically, given any $k < n$, any $h \leqslant r-1$, and a finite field F of cardinality $|F| \geqslant 2n$, we present an $(n, k, l = (h+1)^m)$ MDS array code $\mathcal{C} = \mathcal{C}_{h,k+1}$ over the field F, where $m := \binom{n}{h}$. The code $\mathcal{C}$ has the property that for *any* h-subset $\mathcal{F}$ of $[n]$, the repair of each failed node $C_i, i \in \mathcal{F}$ can be accomplished by connecting to *any* $k+1$ helper nodes and downloading $(k+h)l/(h+1)$ symbols of F in total from these helper nodes as well as from other failed nodes. Clearly, the amount of downloaded data meets the cut-set bound (3) with equality.

As in the previous sections, we will define $\mathcal{C}$ by its parity-check equations, and we begin with some notation. Let $\{\lambda_{ij}\}_{i \in [n], j \in \{0,1\}}$ be $2n$ distinct elements of the field F. Let g be a bijection between the set of h-subsets $\{\mathcal{F} : \mathcal{F} \subseteq [n], |\mathcal{F}| = h\}$ and the numbers $\{1, 2, \dots, m\}$. As in (19), the particular choice of g does not matter; for instance, we can take

$$\begin{gathered}g(\{i_h, i_{h-1}, \dots, i_1\}) = \sum_{j=0}^{h-1} \binom{i_{h-j} - 1}{h-j} + 1\\ \text{for all } n \geqslant i_h > i_{h-1} > \cdots > i_1 \geqslant 1,\end{gathered} \tag{36}$$

where we use the convention that $\binom{n_1}{n_2} = 0$ if $n_1 < n_2$. For a given $a = 0, 1, 2, \dots, l-1$, let $a_m, a_{m-1}, \dots, a_1$ be the digits of its expansion in the base $h+1$, i.e., $a = \sum_{j=0}^{m-1} a_{j+1} (h+1)^j$. For a set $\mathcal{F} \subseteq [n]$ and an element $i \in \mathcal{F}$, let $z(\mathcal{F}, i) = |\{j : j \in \mathcal{F}, j \leqslant i\}|$ be the number of elements in $\mathcal{F}$ that are no larger than i. Define the following function:

$$\begin{aligned}f &: [n] \times \{0, 1, \dots, l-1\} \to \{0, 1\}\\ &(i, a) \mapsto \Big(\sum_{\substack{\mathcal{F} \subseteq [n], |\mathcal{F}| = h\\ \mathcal{F} \ni i}} \mathbb{1}\{a_{g(\mathcal{F})} = z(\mathcal{F}, i)\} \Big) \pmod 2,\end{aligned} \tag{37}$$

where $\mathbb{1}(\cdot)$ is the indicator function. Finally, given $a = 0, 1, \dots, l-1$, $i \in [m]$ and $u = 0, 1, 2, \dots, h$, let $a(i, u) := (a_m, \dots, a_{i+1}, u, a_{i-1}, \dots, a_1)$.

Definition 6. *The code $\mathcal{C} = \mathcal{C}_{h,k+1}$ is defined by the following rl parity-check equations:*

$$\begin{gathered}\sum_{i=1}^{n} \lambda_{i, f(i,a)}^t c_{i,a} = 0,\\ t = 0, 1, 2, \dots, r-1; \; a = 0, 1, 2, \dots, l-1.\end{gathered}$$

For a given $a = 0, 1, 2, \dots, l-1$ the vectors $(c_{1,a}, c_{2,a}, \dots, c_{n,a})$ form an (n, k) MDS code. Therefore $\mathcal{C}$ is indeed an (n, k, l) MDS array code.

Let us show that $\mathcal{C}$ has the $(h, k+1)$-optimal repair property. As before, we define sums of particular entries of the ith node. Namely, let $\mathcal{F} = \{i_1, i_2, \dots, i_h\}$, where $i_1 < i_2 < \cdots < i_h$, be an h-subset of $[n]$. Given $a = 0, 1, \dots, l-1, j \in [h]$ and $i \in [n]$, let

$$\mu_{i,i_j}^{(a)} := c_{i, a(g(\mathcal{F}), 0)} + c_{i, a(g(\mathcal{F}), j)}.$$

The following lemma implies the optimal bandwidth of $\mathcal{C}$ for repairing h failed nodes.

Lemma 8. *Let $\mathcal{F}=\{i_1,i_2,\dots,i_h\}$ be the set of failed nodes. For any set of helper nodes $\mathcal{R}\subseteq[n]\backslash\mathcal{F},|\mathcal{R}|=k+1$, any $j\in[h]$, and any $a\in\{0,1,\dots,l-1\}$, the values of $c_{i_j,a(g(\mathcal{F}),0)},c_{i_j,a(g(\mathcal{F}),j)}$ and $\{\mu_{i,i_j}^{(a)}:i\in\mathcal{F}\backslash\{i_j\}\}$ are uniquely determined by $\{\mu_{i,i_j}^{(a)}:i\in\mathcal{R}\}$.*

The proof of this lemma relies on the same ideas as the proofs of Lemmas 4 and 6. For completeness we outline it at the end of this section.

Let us explain why Lemma 8 implies that $C_i,i\in\mathcal{F}$ can be repaired with optimal bandwidth. In the first round of the repair process, every failed node $C_{i_j},j\in[h]$ downloads $\{\mu_{i,i_j}^{(a)}:a_{g(\mathcal{F})}=0\}$ from each helper node $C_i,i\in\mathcal{R}$. According to Lemma 8, after the first round, C_{i_j} knows the values of

$$\begin{aligned}\{c_{i_j,a}:&a_{g(\mathcal{F})}=0\}\cup\{c_{i_j,a(g(\mathcal{F}),j)}:a_{g(\mathcal{F})}=0\}\\&\cup\{c_{i,a}+c_{i,a(g(\mathcal{F}),j)}:a_{g(\mathcal{F})}=0,i\in\mathcal{F}\backslash\{i_j\}\}.\end{aligned}$$

In the second round of the repair process, every failed node $C_{i_j},j\in[h]$ downloads $\{c_{i_j,a}+c_{i_j,a(g(\mathcal{F}),j')}:a_{g(\mathcal{F})}=0\}$ from each of the other failed nodes $C_{i_{j'}},j'\in[h]\backslash\{j\}$. As a result, C_{i_j} knows the values of all the elements in the set

$$\begin{aligned}\{c_{i_j,a(g(\mathcal{F}),u)}:&a_{g(\mathcal{F})}=0,u=0,1,\dots,h\}\\=&\{c_{i_j,a}:a\in\{0,1,2,\dots,l-1\}\},\end{aligned}$$

or, in other words, C_{i_j} can recover all its coordinates. In regards to the repair bandwidth expended during the two rounds of communication, every failed node $C_{i_j},j\in[h]$ downloads $l/(h+1)$ symbols of F from each of the nodes $C_i,i\in\mathcal{R}\cup\mathcal{F}\backslash\{i_j\}$. Therefore the total repair bandwidth is $h(k+h)l/(h+1)$, meeting the cut-set bound (3) with equality.

Proof of Lemma 8: The parity-check equations that correspond to the rows labeled by $a(g(\mathcal{F}),0)$, $a(g(\mathcal{F}),1),\dots,a(g(\mathcal{F}),h)$ are as follows:

$$\begin{gathered}\sum_{i=1}^n\lambda_{i,f(i,a(g(\mathcal{F}),u))}^t c_{i,a(g(\mathcal{F}),u)}=0,\\ t=0,1,2,\dots,r-1,\ u=0,1,2,\dots,h.\end{gathered}\tag{38}$$

According to definition of the function f in (37), if $i\notin\mathcal{F}$, then the value of $f(i,a)$ does not depend on the digit of a in position $g(\mathcal{F})$. Thus we have

$$\begin{aligned}f(i,a(g(\mathcal{F}),0))&=f(i,a(g(\mathcal{F}),1))\\&=\dots=f(i,a(g(\mathcal{F}),h)),i\in[n]\backslash\mathcal{F}.\end{aligned}$$

Likewise we have for any $j\in[h]$

$$\begin{aligned}&f(i_j,a(g(\mathcal{F}),0))\neq f(i_j,a(g(\mathcal{F}),j)),\\&f(i_j,a(g(\mathcal{F}),0))=f(i_j,a(g(\mathcal{F}),j')),\ j'\in[h]\backslash\{j\}.\end{aligned}$$

Thus we are justified in using the following notation:

$$\begin{aligned}\lambda_i:=\lambda_{i,f(i,a(g(\mathcal{F}),0))}&=\lambda_{i,f(i,a(g(\mathcal{F}),1))}\\&=\dots=\lambda_{i,f(i,a(g(\mathcal{F}),h))},\ i\in[n]\backslash\mathcal{F};\end{aligned}\tag{39}$$

$$\begin{aligned}\lambda_{i_j,0}':=\lambda_{i_j,f(i_j,a(g(\mathcal{F}),0))}&=\lambda_{i_j,f(i_j,a(g(\mathcal{F}),j'))},\\&j\in[h],\ j'\in[h]\backslash\{j\};\end{aligned}\tag{40}$$

$$\lambda_{i_j,1}':=\lambda_{i_j,f(i_j,a(g(\mathcal{F}),j))},\ j\in[h].$$

Notice that

$$\lambda_{i_j,0}'\neq\lambda_{i_j,1}'$$

and

$$\begin{gathered}\{\lambda_{i_j,0}',\lambda_{i_j,1}'\}=\{\lambda_{i_j,0},\lambda_{i_j,1}\}\text{ for all }j\in[h],\\ \lambda_i\in\{\lambda_{i,0},\lambda_{i,1}\},\ i\in[n]\backslash\mathcal{F}.\end{gathered}$$

Therefore the elements $\lambda_{i_1,0}',\lambda_{i_2,0}',\dots,\lambda_{i_h,0}',\lambda_{i_1,1}',\lambda_{i_2,1}',\dots,\lambda_{i_h,1}',\lambda_i,i\in[n]\backslash\mathcal{F}$ are all distinct. Now we can write (38) as

$$\begin{gathered}\sum_{j=1}^h(\lambda_{i_j,0}')^t c_{i_j,a(g(\mathcal{F}),0)}+\sum_{i\in[n]\backslash\mathcal{F}}\lambda_i^t c_{i,a(g(\mathcal{F}),0)}=0,\\ t=0,1,\dots,r-1;\end{gathered}$$

$$\begin{aligned}(\lambda_{i_u,1}')^t c_{i_u,a(g(\mathcal{F}),u)}&+\sum_{j\in[h]\backslash\{u\}}(\lambda_{i_j,0}')^t c_{i_j,a(g(\mathcal{F}),u)}\\&+\sum_{i\in[n]\backslash\mathcal{F}}\lambda_i^t c_{i,a(g(\mathcal{F}),u)}=0,\\ t=0,1,&\dots,r-1,\ u=1,2,\dots,h.\end{aligned}$$

These equations have exactly the same form as the equations in (35). Therefore the remainder of the proof of Lemma 8 follows the steps in the proof of Lemma 7 (or Lemma 3), and we do not repeat them here.

VII. Cooperative (h,d)-optimal codes for general h and general d

A. Repairing the first h nodes from any d helper nodes

In this section we present a construction of MDS array codes that can optimally repair the first h nodes from any $d\geqslant k+1$ helper nodes for any given $2\leqslant h\leqslant n-d\leqslant r-1$. (We do not consider the case of $d=k$ because codes for it were constructed earlier in [16].) Let $s:=d+1-k$. Given a finite field F of cardinality $|F|\geqslant n+h(s-1)$, we present an $(n,k,l=(h+s-1)(s-1)^{h-1})$ MDS array code $\mathcal{C}=\mathcal{C}_{h,d}^{(0)}$ over the field F that has the following property: When the first h nodes of $\mathcal{C}$ fail, the repair of each failed node can be accomplished by connecting to *any* d helper nodes and downloading

$$(d+h-1)\frac{l}{d+h-k}=(d+h-1)(s-1)^{h-1}$$

symbols of F in total from these helper nodes as well as from the other failed nodes. Clearly, the amount of downloaded data meets the cut-set bound (3) with equality.

Let $(\lambda_{ij},i=1,\dots,h,j=0,1,\dots,s-1),\lambda_{h+1},\lambda_{h+2},\dots,\lambda_n$ be $hs+n-h$ distinct elements of the field F. Define

$$\begin{aligned}A:=\Big\{\underline{a}=(a_1,a_2,\dots,a_h):\underline{a}\in\{0,1,\dots,s-1\}^h,&\\ \sum_{i=1}^h\mathbb{1}\{a_i=s-1\}\leqslant1\Big\}&,\end{aligned}\tag{41}$$

i.e., A is the subset of $\{0,1,\dots,s-1\}^h$ consisting of all the $\underline{a}$ such that at most one of its coordinates is $s-1$. It is easy to verify that

$$|A|=(h+s-1)(s-1)^{h-1}=l.\tag{42}$$

Let $C = (C_1, C_2, \dots, C_n) \in \mathcal{C}$ be a codeword of the code $\mathcal{C}$. In this section, we use a multi-index (vector) notation $\underline{a} = (a_1, a_2, \dots, a_h)$ to label the entries of each node C_i, so the node has the form $C_i = (c_{i,\underline{a}}, \underline{a} \in A)$. In previous sections we opted for numbering the entries of C_i with integers even though on several occasions (e.g., in Sections IV-B, V-B) we have essentially relied on the multi-index notation. We could follow this pattern in this section as well, however the integer numbering would not be consecutive, and we find the vector notation much more convenient for the presentation. We note that, according to (42), the dimension of C_i over F is indeed l.

Definition 7. *The code $\mathcal{C}$ is defined by the following parity check equations:*

$$\sum_{i=1}^{h} \lambda_{i,a_i}^t c_{i,\underline{a}} + \sum_{i=h+1}^{n} \lambda_i^t c_{i,\underline{a}} = 0, \; t = 0, 1, \dots, r-1, \; \underline{a} \in A. \tag{43}$$

Since for each $\underline{a} \in A$, the set of vectors $\{(c_{1,\underline{a}}, c_{2,\underline{a}}, \dots, c_{n,\underline{a}})\}$ forms an (n, k) MDS code, $\mathcal{C}$ is indeed an (n, k, l) MDS array code.

1) Intuition behind the repair scheme: We begin with an informal discussion of the code construction and the accompanying repair scheme. According to the cut-set bound (3), if we assume that the amount of communication between any two nodes is the same (*uniform download*), which is the case for our repair scheme, then this amount is equal to $\frac{l}{h+d-k} = (s-1)^{h-1}$ symbols of F. More precisely, in the first round of repair process, each failed node should download $(s-1)^{h-1}$ symbols of F from each helper node, and in the second round, each failed node should download $(s-1)^{h-1}$ symbols of F from each of the other failed nodes.

For $i \in [h]$ and $u \in \{0, 1, \dots, s-1\}$, define $\underline{a}(i, u) := (a_1, a_2, \dots, a_{i-1}, u, a_{i+1}, a_{i+2}, \dots, a_h)$. For $i \in [h]$, define the set of indices

$$B_i := \big\{\underline{a} = (a_1, a_2, \dots, a_h) : a_i \in [0, s-1], \\ a_j \in [0, s-2] \text{ for all } j \neq i\big\},$$

where $[0, t] := \{0, 1, \dots, t\}$ for an integer t. Define $A_0 := \{0, 1, \dots, s-2\}^h$. It is easy to see that

$$\bigcup_{i=1}^{h} B_i = A, \quad \bigcap_{i=1}^{h} B_i = A_0.$$

In the first round of repair, each failed node $C_i, i \in [h]$ connects to d helper nodes $C_j, j \in \mathcal{R}$ and downloads $(s-1)^{h-1}$ symbols from each of them, so altogether it acquires $d(s-1)^{h-1}$ symbols of F. This enables C_i to recover a certain portion of its entries, which we can quantify relying on the cut-set bound. For this, we observe that this bound gives a lower estimate on the repair bandwidth for a given size of each node l. At the same time, given the repair bandwidth, it gives an upper estimate on the node size, including in particular a bound on the maximum number of entires of the node that can be recovered from a certain amount of the downloaded data. Using this observation, let us take $|\mathcal{F}| = 1$ and $|\mathcal{R}| = d$ in (2) (or in (3)), and replace the left-hand side with $d(s-1)^{h-1}$. Solving for l, we see that each failed node can recover at most $s(s-1)^{h-1}$ coordinates. At the same time, the cardinality of the set B_i is exactly $s(s-1)^{h-1}$, and this is the subset of the entries of C_i that will be repaired after the first round of communication. Namely, according to Lemma 2, the set of values $\{c_{i,\underline{a}} : \underline{a} \in B_i\}$ can be found relying on the values

$$\Big\{\Big(\sum_{u=0}^{s-1} c_{j,\underline{a}(i,u)} : \underline{a} \in B_i, a_i = 0\Big), j \in \mathcal{R}\Big\}$$

(see Lemma 9 below), and therefore, the node C_i downloads the set $\{\sum_{u=0}^{s-1} c_{j,\underline{a}(i,u)} : \underline{a} \in B_i, a_i = 0\}$ from each of the helper nodes $C_j, j \in \mathcal{R}$. Since for every $\underline{a} \in B_i$ the coordinate a_i can take s possible values, the number of symbols downloaded from each of them is exactly $\frac{|B_i|}{s} = (s-1)^{h-1}$.

To move forward, we note that Lemma 2 gives us more: namely, apart from the values $\{c_{i,\underline{a}} : \underline{a} \in B_i\}$, each $C_i, i \in [h]$ can also compute $(s-1)^{h-1}$ *sums of coordinates of the other failed nodes*. Namely, after the first round, C_i can find the values

$$\Big\{\sum_{u=0}^{s-1} c_{j,\underline{a}(i,u)} : \underline{a} \in B_i, a_i = 0\Big\} \quad \text{for all } j \in [h]\backslash\{i\}. \tag{44}$$

This is the information that will be exchanged between the failed nodes $C_i, i \in [h]$ in the second round.

To describe the second part of the repair scheme, we note that the number of coordinates still not available at the node C_i equals

$$|A\backslash B_i| = l - s(s-1)^{h-1} = (h-1)(s-1)^{h-1}.$$

As noted above (again assuming uniform download), in the second round each failed node should download $(s-1)^{h-1}$ symbols of F from each of the other $(h-1)$ failed nodes. Therefore, in the second round, each failed node should acquire $(h-1)(s-1)^{h-1}$ symbols of F, which matches the number of the still missing symbols of the node. To decide what to download we turn to (44), noting that each failed node C_i knows the sums in (44) for all the other failed nodes $C_j, j \in [h]\backslash\{i\}$. For a fixed j, there are $(s-1)^{h-1}$ symbols in the set (44), so a natural thing to do in the second round is to let C_i transmit the sums in (44) to each of the remaining failed nodes $C_j, j \in [h]\backslash\{i\}$.

Since every failed node C_j knows $\{c_{j,\underline{a}} : \underline{a} \in B_j\}$ after the first round and $A_0 \subset B_j$ for all $j \in [h]$, every failed node C_j knows $\{c_{j,\underline{a}} : \underline{a} \in A_0\}$. We observe that each sum in (44) has s terms and that the indices of $s-1$ of them belong to the set A_0, so C_j can calculate the single remaining term from each of these sums. Upon completing this calculation, the node C_j knows the values of all the summands of all the sums in the set (44), i.e., C_j knows all the coordinates in the set $\{c_{j,\underline{a}} : \underline{a} \in B_i\}$. Since C_j downloads these sums from all the other failed nodes $C_i, i \in [h]\backslash\{j\}$, the downloaded symbols in the second round enable C_j to calculate the coordinates

$$\bigcup_{i \in [h]\backslash\{j\}} \{c_{j,\underline{a}} : \underline{a} \in B_i\}.$$

Recall that after the first round, C_j already knows the values of coordinates $\{c_{j,\underline{a}} : \underline{a} \in B_j\}$. Thus after the whole repair process, C_j can find the entries

$$\Big\{c_{j,\underline{a}} : \underline{a} \in \bigcup_{i=1}^{h} B_i\Big\} = \{c_{j,\underline{a}} : \underline{a} \in A\}.$$

This concludes the repair procedure because C_j has found all the missing l entries.

2) Formal description and validity proof of the repair scheme: The discussion in the previous subsection contains most of what is needed to justify the repair scheme. The omitted step is a connection with Lemma 2 which we include next.

Lemma 9. *Let $C_i, i \in [h]$ be one of the failed nodes, and let $\mathcal{R} \subseteq [n]\backslash[h]$ be the indices of helper nodes, where $|\mathcal{R}| = d$. For any $\underline{a} \in B_i$, the elements $c_{i,\underline{a}(i,0)}, c_{i,\underline{a}(i,1)}, \dots, c_{i,\underline{a}(i,s-1)}$ and the values of $\{\sum_{u=0}^{s-1} c_{j,\underline{a}(i,u)} : j \in [h]\backslash\{i\}\}$ can be calculated from the values in the set $\{\sum_{u=0}^{s-1} c_{j,\underline{a}(i,u)} : j \in \mathcal{R}\}$.*

Proof: We again use Lemma 2. Let us write out the parity-check equations (43) that correspond to the indices $\underline{a}(i,0), \underline{a}(i,1), \dots, \underline{a}(i,s-1)$:

$$\lambda_{i,u}^t c_{i,\underline{a}(i,u)} + \sum_{j\in[h]\backslash\{i\}} \lambda_{j,a_j}^t c_{j,\underline{a}(i,u)} + \sum_{j=h+1}^{n} \lambda_j^t c_{j,\underline{a}(i,u)} = 0,$$
$$t = 0,1,\dots,r-1, \quad u = 0,1,\dots,s-1. \tag{45}$$

We can see that this set of equations has the same form as (14): In (45) only the coefficients of $c_{i,\underline{a}(i,u)}$ vary with u while the coefficients of $c_{j,\underline{a}(i,u)}$ are independent of u for all $j \in [n]\backslash\{i\}$; in (14) only the coefficients of $c_{1,u}$ vary with u while the coefficients of $c_{j,u}$ are independent of u for all $j \in [n]\backslash\{1\}$. Therefore Lemma 2 applies directly, and the proof is complete. ■

In the first round, each failed node $C_i, i \in [h]$ downloads

$$\Big\{ \sum_{u=0}^{s-1} c_{j,\underline{a}(i,u)} : \underline{a} \in B_i, a_i = 0 \Big\} \tag{46}$$

from each helper node $C_j, j \in \mathcal{R}$. As already explained, the cardinality of the set in (46) is $(s-1)^{h-1}$.

According to Lemma 9, after the first round, each failed node $C_i, i \in [h]$ knows the following field elements:

$$\{c_{i,\underline{a}} : \underline{a} \in B_i\} \bigcup \Big(\bigcup_{j\in[h]\backslash\{i\}} \Big\{ \sum_{u=0}^{s-1} c_{j,\underline{a}(i,u)} : \underline{a} \in B_i, a_i = 0 \Big\} \Big).$$

In the second round, each failed node $C_j, j \in [h]$ downloads

$$\Big\{ \sum_{u=0}^{s-1} c_{j,\underline{a}(i,u)} : \underline{a} \in B_i, a_i = 0 \Big\}$$

from each of the other failed nodes $C_i, i \in [h]\backslash\{j\}$. According to the arguments above, after the second round each failed node can recover all its coordinates, and the repair bandwidth achieves the cut-set bound (3) with equality.

3) Connections with $\mathcal{C}_{2,d}^{(0)}$ and $\mathcal{C}_{h,k+1}^{(0)}$: Let us look back at the codes $\mathcal{C}_{2,d}^{(0)}$ and $\mathcal{C}_{h,k+1}^{(0)}$ which are special cases of the above construction (although this may be not immediate to see, which justifies their independent description earlier in the paper). Namely, the code $\mathcal{C}_{h,d}^{(0)}$ with $h = 2$ becomes the same as $\mathcal{C}_{2,d}^{(0)}$, albeit with a different way of indexing the entries of each node C_i, and similarly, letting $d = k+1$ in $\mathcal{C}_{h,d}^{(0)}$, we obtain the code $\mathcal{C}_{h,k+1}^{(0)}$ with a different way of indexing.

First, using Table I, it is immediate to see that the subpacketization values match. Now let us verify the easier of the two specializations, checking the case of $h = 2$. Indeed, in this case the set A defined in (41) becomes

$$A = \{\underline{a} = (a_1, a_2) : a_1, a_2 \in \{0, 1, \dots, s-1\},$$
$$(a_1, a_2) \neq (s-1, s-1)\}.$$

A natural way to transform the multi-index $\underline{a} = (a_1, a_2)$ into an integer index is to use the mapping $a = a_1 + s a_2$. It is clear that the image of A under this mapping is $\{0, 1, 2, \dots, s^2 - 2\}$, which is exactly the same as the set of integer indices in Section V-A. One can further check that when $h = 2$, the parity check equations of $\mathcal{C}_{h,d}^{(0)}$ given in (43) are the same as the parity check equations (25) of $\mathcal{C}_{2,d}^{(0)}$.

Let us now explain that using $d = k+1$ in the description of the code $\mathcal{C}_{h,d}^{(0)}$, we obtain $\mathcal{C}_{h,k+1}^{(0)}$. When $d = k+1$, the set A defined in (41) becomes

$$A = \{\underline{0}, e_1, e_2, \dots, e_h\},$$

where $\underline{0}$ is an all-zero vector of length h, and for $i \in [h]$, e_i is the h-dimensional vector whose only nonzero coordinate is located at the ith position, and this coordinate is 1. We map $\underline{0}$ to 0 and e_i to i for all $i \in [h]$. It is easy to check that under this mapping the parity-check equations (43) of the code $\mathcal{C}_{h,d}^{(0)}$ are the same as the parity-check equations (35) of $\mathcal{C}_{h,k+1}^{(0)}$.

B. Repairing any h nodes from any d helper nodes

Finally, in this section we present the codes $\mathcal{C} = \mathcal{C}_{h,d}$ that address the most general case of the repair problem. As above, we let $s := d + 1 - k$ and suppose that $F, |F| \geqslant sn$ is a finite field. We present an $(n, k, l = ((h+s-1)(s-1)^{h-1})^m)$ MDS array code $\mathcal{C} = \mathcal{C}_{h,d}$ over F, where $m := \binom{n}{h}$. The code $\mathcal{C}$ has the property that for *any* h-subset $\mathcal{F}$ of $[n]$, the repair of each failed node $C_i, i \in \mathcal{F}$ can be accomplished by connecting to *any* d helper nodes and downloading $(d+h-1)l/(h+s-1)$ symbols of F in total from these helper nodes as well as from the other failed nodes. Clearly, the amount of downloaded data meets the cut-set bound (3) with equality, and so the code $\mathcal{C}$ supports optimal repair.

Let $\{\lambda_{ij}, i = 1, \dots, n, j = 0, 1, \dots, s-1\}$ be sn distinct elements of the field F. We will rely on the definition of the set A in (41). To remind ourselves, this is the set of h-tuples of integers between 0 and $s-1$ that contain at most one entry equal to $s-1$. We use the shorthand notation $[0, i] := \{0, 1, \dots, i\}$ for an integer i, and define a set of integer vectors $A^{[m]} \subset [0, s-1]^{hm}$ such that each of the m subvectors is contained in A. More specifically, in this section we use $\underline{a}$ to denote an integer vector of length hm:

$$\underline{a} = (\underline{a}^{(1)}, \underline{a}^{(2)}, \dots, \underline{a}^{(m)}), \tag{47}$$

where $\underline{a}^{(i)} = (a_1^{(i)}, \dots, a_h^{(i)}) \in [0, s-1]^h$. Define the set

$$A^{[m]} := \{\underline{a} \in [0, s-1]^{hm} : \underline{a}^{(i)} \in A, i = 1, \dots, m\}.$$

According to (42), each $\underline{a}^{(i)}$ can take $(h+s-1)(s-1)^{h-1}$ possible values, so

$$\big|A^{[m]}\big| = \big((h+s-1)(s-1)^{h-1}\big)^m = l. \tag{48}$$

Let g be the bijection between the set of h-subsets $\{\mathcal{F} : \mathcal{F} \subseteq [n], |\mathcal{F}| = h\}$ and the numbers $\{1, 2, \dots, m\}$ defined in (36). For a set $\mathcal{F} \subseteq [n]$ and an element $i \in \mathcal{F}$, let $z(\mathcal{F}, i) = |\{j : j \in \mathcal{F}, j \leqslant i\}|$ be the number of elements in $\mathcal{F}$ that are not greater than i. Define the following function:

$$\begin{aligned} f : [n] \times A^{[m]} &\to \{0, 1, \dots, s-1\} \\ (i, \underline{a}) &\mapsto \Bigg(\sum_{\substack{\mathcal{F} \subseteq [n], |\mathcal{F}| = h \\ \mathcal{F} \ni i}} \underline{a}_{z(\mathcal{F}, i)}^{(g(\mathcal{F}))} \Bigg) \pmod{s}, \end{aligned} \tag{49}$$

Let $C = (C_1, C_2, \dots, C_n) \in \mathcal{C}$ be a codeword of the code $\mathcal{C}$. We index the entries of the code C_i using the multi-index $\underline{a}$ defined above in (47), writing $C_i = (c_{i,\underline{a}}, \underline{a} \in A^{[m]})$. According to (48), the dimension of C_i over F is indeed l. The last element of notation is as follows: for every $\underline{a} \in A^{[m]}$, $i \in [m]$ and $\underline{b} \in A$, let

$$\underline{a}(i, \underline{b}) := (\underline{a}^{(1)}, \underline{a}^{(2)}, \dots, \underline{a}^{(i-1)}, \underline{b}, \underline{a}^{(i+1)}, \dots \underline{a}^{(m)}).$$

Definition 8. *The code $\mathcal{C} = \mathcal{C}_{h,d}$ is defined by the following rl parity-check equations:*

$$\sum_{i=1}^{n} \lambda_{i, f(i, \underline{a})}^{t} c_{i, \underline{a}} = 0, \ t = 0, 1, 2, \dots, r-1; \ \underline{a} \in A^{[m]}. \tag{50}$$

For every $\underline{a} \in A^{[m]}$, the vectors $(c_{1,\underline{a}}, c_{2,\underline{a}}, \dots, c_{n,\underline{a}})$ form an (n, k) MDS code. Therefore $\mathcal{C}$ is indeed an (n, k, l) MDS array code.

Let us show that $\mathcal{C}$ has the (h, d)-optimal repair property. Let $\mathcal{F} = \{i_1, i_2, \dots, i_h\}$, where $1 \leqslant i_1 < i_2 < \cdots < i_h \leqslant n$, be the set of indices of h failed nodes. For every codeword $C = (C_1, C_2, \dots, C_n) \in \mathcal{C}$ and every $\underline{a} \in A^{[m]}$, we form a vector $C^{(\underline{a})}$ by taking a subset of coordinates from each node $C_i, i \in [n]$:

$$C^{(\underline{a})} := (C_1^{(\underline{a})}, C_2^{(\underline{a})}, \dots, C_n^{(\underline{a})}),$$

where

$$C_i^{(\underline{a})} := (c_{i, \underline{a}(g(\mathcal{F}), \underline{b})} : \underline{b} \in A), \quad i = 1, \dots, n. \tag{51}$$

By definition the set $C_i^{(\underline{a})}$ contains $(h+s-1)(s-1)^{h-1}$ coordinates of C_i. Since the indices of these coordinates are obtained by replacing the subvector $\underline{a}^{(g(\mathcal{F}))}$ with all the vectors of the set A, the vectors $C^{(\underline{a})}$ and $C_i^{(\underline{a})}$ do not depend on the original value of $\underline{a}^{(g(\mathcal{F}))}$, i.e.,

$$\begin{aligned} C^{(\underline{a})} = C^{(\underline{a}(g(\mathcal{F}), \underline{b}))} \text{ and } C_i^{(\underline{a})} = C_i^{(\underline{a}(g(\mathcal{F}), \underline{b}))} \\ \text{for all } C \in \mathcal{C}, i \in [n] \text{ and } \underline{b} \in A. \end{aligned} \tag{52}$$

Moreover, consider the following $((h+s-1)(s-1)^{h-1})^{m-1}$ sets of coordinates of C_i:

$$\{C_i^{(\underline{a})} : \underline{a} \in A^{[m]}, \underline{a}^{(g(\mathcal{F}))} = \underline{0}\}, \tag{53}$$

where we view each vector $C_i^{(\underline{a})}$ defined in (51) as a set. Since we are limiting the subvector $\underline{a}^{(g(\mathcal{F}))}$ to 0 while originally it can take $|A| = (h+s-1)(s-1)^{h-1}$ values, the vector $\underline{a}$ in (53) takes

$$\frac{l}{(h+s-1)(s-1)^{h-1}} = ((h+s-1)(s-1)^{h-1})^{m-1}$$

possible values. Therefore (53) contains $((h+s-1)(s-1)^{h-1})^{m-1}$ distinct sets of coordinates of C_i. This amounts to saying that the sets in (53) form a partition of the coordinates of C_i.

For every $\underline{a} \in A^{[m]}$, we define an $(n, k, (h+s-1)(s-1)^{h-1})$ MDS array code $\mathcal{C}^{(\underline{a})}$ as follows:

$$\mathcal{C}^{(\underline{a})} := \{(C_1^{(\underline{a})}, C_2^{(\underline{a})}, \dots, C_n^{(\underline{a})}) : C \in \mathcal{C}\},$$

where the MDS property and the dimension of $\mathcal{C}^{(\underline{a})}$ follow directly from the definition of the code $\mathcal{C}$; see (50), (51). To better understand the connection between the code $\mathcal{C}$ and its subcodes $\mathcal{C}^{(\underline{a})}, \underline{a} \in A^{[m]}$, we can view each codeword of $\mathcal{C}$ as a two-dimensional array of size $l \times n$. We use multi-index $\underline{a} \in A^{[m]}$ to index each row and $i \in [n]$ to index each column of the codeword. Each subcode $\mathcal{C}^{(\underline{a})}, \underline{a} \in A^{[m]}$ contains $(h+s-1)(s-1)^{h-1}$ rows of the codewords in $\mathcal{C}$, and the indices of these $(h+s-1)(s-1)^{h-1}$ rows are in the set $\{\underline{a}(g(\mathcal{F}), \underline{b}) : \underline{b} \in A\}$. From (52) it is clear that

$$\mathcal{C}^{(\underline{a})} = \mathcal{C}^{(\underline{a}(g(\mathcal{F}), \underline{b}))} \text{ for all } \underline{b} \in A.$$

Thus, the code $\mathcal{C}$ can be partitioned into $((h+s-1)(s-1)^{h-1})^{m-1}$ subcodes

$$\{\mathcal{C}^{(\underline{a})} : \underline{a} \in A^{[m]}, \underline{a}^{(g(\mathcal{F}))} = \underline{0}\},$$

and each subcode contains $(h+s-1)(s-1)^{h-1}$ rows of the code $\mathcal{C}$. We will show that each of these subcodes has the same structure as the code $\mathcal{C}_{h,d}^{(0)}$ defined in Section VII-A, and can therefore be optimally repaired.

Lemma 10. *For every $\underline{a} \in A^{[m]}$, the $(n, k, (h+s-1)(s-1)^{h-1})$ MDS array code $\mathcal{C}^{(\underline{a})}$ can optimally repair the failed nodes $C_i^{(\underline{a})}, i \in \mathcal{F}$ from any d helper nodes, i.e., the bandwidth of repairing $C_i^{(\underline{a})}, i \in \mathcal{F}$ from any d helper nodes achieves* (3) *with equality.*

Proof: Our goal is to show that the code $\mathcal{C}^{(\underline{a})}$ has the same structure as the code $\mathcal{C}_{h,d}^{(0)}$. Then we can apply the optimal repair scheme for the first h nodes of $\mathcal{C}_{h,d}^{(0)}$ to the repair of the failed nodes of $\mathcal{C}^{(\underline{a})}$ whose indices are in $\mathcal{F}$.

By definition (49), the function f has the following property: For any $\underline{a} \in A^{[m]}$ and any $\underline{b} = (b_1, b_2, \dots, b_h) \in A$,

$$\begin{aligned} f(i, \underline{a}(g(\mathcal{F}), \underline{b})) &= f(i, \underline{a}) \text{ for all } i \in [n] \backslash \mathcal{F}, \\ f(i_u, \underline{a}(g(\mathcal{F}), \underline{b})) &= f(i_u, \underline{a}(g(\mathcal{F}), \underline{0})) \oplus b_u \text{ for all } u \in [h], \end{aligned} \tag{54}$$

where $\underline{0}$ is the all-zero vector of length h, and $\oplus$ is addition modulo s. From now on we fix an $\underline{a} \in A^{[m]}$ and prove the claim for this fixed $\underline{a}$. According to (54), we are justified in using the following notation:

$$\begin{aligned} \lambda_i := \lambda_{i, f(i, \underline{a})} &= \lambda_{i, f(i, \underline{a}(g(\mathcal{F}), \underline{b}))} \\ &\text{for all } i \in [n] \backslash \mathcal{F} \text{ and all } \underline{b} \in A. \end{aligned} \tag{55}$$

We further define

$$\begin{aligned} \lambda'_{i_u, j} :=& \lambda_{i_u, f(i_u, \underline{a}(g(\mathcal{F}), \underline{0})) \oplus j} \\ &\text{for all } u \in [h] \text{ and all } j \in \{0, 1, \dots, s-1\}. \end{aligned}$$

Again by (54), we have

$$\begin{aligned} \lambda'_{i_u, b_u} = \lambda_{i_u, f(i_u, \underline{a}(g(\mathcal{F}), \underline{0})) \oplus b_u} &= \lambda_{i_u, f(i_u, \underline{a}(g(\mathcal{F}), \underline{b}))} \\ &\text{for all } u \in [h] \text{ and all } \underline{b} \in A. \end{aligned} \tag{56}$$

By (51), $C_i^{(\underline{a})}$ consists of the coordinates $(c_{i,\underline{a}(g(\mathcal{F}),\underline{b})} : \underline{b} \in A)$. Using (50), (55) and (56), we can write out the parity check equations of $\mathcal{C}^{(\underline{a})}$ as follows:

$$\sum_{u=1}^{h} (\lambda'_{i_u,b_u})^t c_{i_u,\underline{a}(g(\mathcal{F}),\underline{b})} + \sum_{i\in[n]\backslash\mathcal{F}} \lambda_i^t c_{i,\underline{a}(g(\mathcal{F}),\underline{b})} = 0, \qquad t = 0,1,\dots,r-1, \quad \underline{b} \in A. \tag{57}$$

We can check that (57) has the same form as (43). Indeed, $\underline{b}$ in (57) plays the role of $\underline{a}$ in (43); the first sum in both equations consists of coordinates of the h failed nodes, and the second sum in both equations consists of coordinates of the other available nodes; in both equations, only the coefficients of the coordinates of the failed nodes vary with the indices, and they vary in exactly the same way. Therefore the repair scheme of code $\mathcal{C}_{h,d}^{(0)}$ can be directly applied to the repair of $C_i^{(\underline{a})}, i \in \mathcal{F}$ from any d helper nodes, and the repair bandwidth of this scheme achieves the bound (3). This completes the proof of Lemma 10. ■

Since every subcode can optimally repair the failed nodes whose indices are in the set $\mathcal{F}$, the same is true for the code $\mathcal{C}$: namely it is capable of repairing $C_i, i \in \mathcal{F}$ from any d helper nodes with optimal repair bandwidth.

Remark: Expanding the discussion in Section VII-A3, we can see that both the codes $\mathcal{C}_{2,d}$ and $\mathcal{C}_{h,k+1}$ are special cases of the code $\mathcal{C}_{h,d}$: taking $h = 2$ in the definition of $\mathcal{C}_{h,d}$, we obtain the code $\mathcal{C}_{2,d}$ with a different indexing of the node's coordinates, and in the same way, taking $d = k + 1$ in $\mathcal{C}_{h,d}$, we obtain the code $\mathcal{C}_{h,k+1}$, with a different way of indexing.

C. A family of universal codes

Using the construction in the previous subsection as a building block and exploiting the concatenation operation defined in Section V-C, we can easily construct an (n, k) MDS array code $\mathcal{C}^U$ with universal (h, d)-optimal repair property for all $1 \leqslant h \leqslant n - d \leqslant n - k$ simultaneously. In other words, the codes that we construct can optimally repair any number of erasures from any number of helper nodes.

Indeed, let

$$\mathcal{C}^U := \bigodot_{1\leqslant h\leqslant n-d\leqslant n-k} \mathcal{C}_{h,d}.$$

The code $\mathcal{C}^U$ is simply a concatenation of all $\mathcal{C}_{h,d}$ for $1 \leqslant h \leqslant n - d \leqslant n - k$, where the codes $\mathcal{C}_{h,d}$ for $h \geqslant 2$ are defined in the previous subsection, and the code $\mathcal{C}_{1,d}$ is given in Sec. V-C [4]. It can be constructed over a field F with size $|F| \geqslant rn$, and it supports optimal repair of any single node, and optimal cooperative repair of any $h \geqslant 2$ nodes.

References

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

[2] 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. Inform. Theory*, vol. 57, no. 8, pp. 5227–5239, 2011.

[3] I. Tamo, Z. Wang, and J. Bruck, "Zigzag codes: MDS array codes with optimal rebuilding," *IEEE Trans. Inform. Theory*, vol. 59, no. 3, pp. 1597–1616, 2013.

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

[5] 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.

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

[7] I. Tamo, M. Ye, and A. Barg, "Optimal repair of Reed-Solomon codes: Achieving the cut-set bound," in *Proc. 58th IEEE Sympos. on the Foundations of Computer Science (FOCS), October 15-17, 2017, Berkeley, CA*, pp. 216–227.

[8] M. Blaum, P. G. Farell, and H. van Tilborg, "Array codes," in *Handbook of Coding Theory*, V. Pless and W. C. Huffman, Eds. Elsevier Science, 1998, vol. II, ch. 22, pp. 1855–1909.

[9] 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. Inform. Theory*, vol. 59, no. 5, pp. 2974–2987, 2013.

[10] A. S. Rawat, O. O. Koyluoglu, and S. Vishwanath, "Centralized repair of multiple node failures with applications to communication efficient secret sharing," 2016, arXiv:1603.04822.

[11] Z. Wang, I. Tamo, and J. Bruck, "Optimal rebuilding of multiple erasures in MDS codes," *IEEE Transactions on Information Theory*, vol. 63, no. 2, pp. 1084–1101, 2017.

[12] M. Zorgui and Z. Wang, "Centralized multi-node repair for minimum storage regenerating codes," in *2017 IEEE International Symposium on Information Theory (ISIT)*. IEEE, 2017, pp. 2213–2217.

[13] M. Ye and A. Barg, "Repairing Reed-Solomon codes: Universally achieving the cut-set bound for any number of erasures," 2017, arXiv:1710.07216.

[14] M. Zorgui and Z. Wang, "On the achievability region of regenerating codes for multiple erasures," 2018, arXiv:1802.00104.

[15] A. M. Kermarrec, N. Le Scouarnec, and G. Straub, "Repairing multiple failures with coordinated and adaptive regenerating codes," in *2011 International Symposium on Network Coding (NetCod)*. IEEE, 2011, pp. 1–6.

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

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

[18] K. W. Shum and J. Chen, "Cooperative repair of multiple node failures in distributed storage systems," *International Journal of Information and Coding Theory*, vol. 3, no. 4, pp. 299–323, 2016.

[19] S. Goparaju, A. Fazeli, and A. Vardy, "Minimum storage regenerating codes for all parameters," *IEEE Trans. Inform. Theory*, vol. 63, no. 10, pp. 6318–6328, 2017.

[20] M. Elyasi and S. Mohajer, "Determinant coding: A novel framework for exact-repair regenerating codes," *IEEE Trans. Inform. Theory*, vol. 62, no. 12, pp. 6683–6697, 2016.

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

[22] H. Dau and O. Milenkovic, "Optimal repair schemes for some families of full-length Reed-Solomon codes," in *Proc. 2017 IEEE International Symposium on Information Theory (ISIT)*. IEEE, 2017, pp. 346–350.

[23] H. Dau, I. Duursma, H. M. Kiah, and O. Milenkovic, "Repairing Reed-Solomon codes with multiple erasures," 2016, arXiv:1612.01361.

[24] A. Chowdhury and A. Vardy, "Improved schemes for asymptotically optimal repair of MDS codes," arXiv:1710.01867.

[25] B. Bartan and M. Wootters, "Repairing multiple failures for scalar MDS codes," 2017, arXiv:1707.02241.

[26] M. Ye and A. Barg, "Explicit constructions of MDS array codes and RS codes with optimal repair bandwidth," in *Proc. 2016 IEEE International Symposium on Information Theory (ISIT)*. IEEE, 2016, pp. 1202–1206.

[27] I. Tamo, M. Ye, and A. Barg, "The repair problem for Reed-Solomon codes: Optimal repair of single and multiple erasures, asymptotically optimal node size," 2018, arXiv:1805.01883.

[28] V. R. Cadambe, C. Huang, and J. Li, "Permutation code: Optimal exact-repair of a single failed node in MDS code based distributed storage systems," in *Proc. 2011 IEEE International Symposium on Information Theory (ISIT)*, 2011, pp. 1225–1229.

[29] S. Benedetto, D. Divsalar, G. Montorsi, and F. Pollara, "Serial concatenation of interleaved codes: performance analysis, design, and iterative decoding," *IEEE Trans. Inform. Theory*, vol. 44, no. 3, pp. 909–926, 1998.

Min Ye received the B.S. degree in Electrical Engineering from Peking University, Beijing, China in 2012, and the Ph.D. degree in the Department of Electrical and Computer Engineering, University of Maryland, College Park in 2017. His research interests include coding theory and information theory.

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.