Optimal MDS codes for cooperative repair

Min Ye  Alexander Barg

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. We present two families of \((n, k)\) MDS codes with optimal cooperative repair. Codes in the first family support optimal repair of any two erasures from any \(d\) helper nodes for any given \(k \leq d \leq n-2\). Codes in the second family support optimal repair of any \(h\) failed nodes, \(2 \leq h \leq n-k-1\), from any \(k+1\) helper nodes.

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. \cite{Dimakis}. In paper \cite{Dimakis} 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 \cite{Dimakis} also gave a lower bound on the minimum repair bandwidth of MDS codes, known as the cutset 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 \cite{Gopalan,Shahak,Ye1,Ye2,Ye3}.

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 recovery 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 recovery procedure.

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 collector is responsible for the repair of all the failed nodes \cite{Gopalan,Ye1,Ye2,Ye3}, and the other is the cooperative model, where the failed nodes may communicate but are distinct, and the amount of data exchanged between

---

M. Ye is with Department of EE, 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 RAS, Moscow, Russia, email: abarg@umd.edu. Research partially supported by NSF grants CCF1422955 and CCF1618603.
them is included in the repair bandwidth \[13\]–\[15\]. The cut-set bounds on the repair bandwidth for multiple erasures under these two models were derived in \[8\] and \[14\] respectively.

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 \[16\]. 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.

Let \(\mathcal{F} \subseteq [n], |\mathcal{F}| = h\) 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 data collector 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 collector, 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 process of performing the repair. 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 \(C\) over a finite field \(F\) and let \(C \in \mathcal{C}\) be a codeword. We write \(C\) as \((C_1, C_2, \ldots, C_n)\), where \(C_i = (c_{i,0}, c_{i,1}, \ldots, c_{i,l-1})^T \in F^l, i = 1, \ldots, n\) is the \(i\)th coordinate of \(C\). The node repair models can be formalized as follows.

Definition 1.1 (Centralized model): Let \(\mathcal{F}\) and \(\mathcal{R}\) be the sets of failed and helper nodes, and suppose that \(|\mathcal{F}| = h \leq r\) and \(|\mathcal{R}| = d \leq k\). 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 \(\beta_{\mathcal{F}}(\mathcal{R})\) symbols of \(F\) if for every \(C \in \mathcal{C}\), there are \(d\) numbers \(\beta_j, j \in \mathcal{R}\), \(d\) functions \(f_j : F^l \rightarrow F^{\beta_j}, j \in \mathcal{R}\), and \(h\) functions \(g_i : F^{\sum_{j \in \mathcal{R}} \beta_j} \rightarrow F^l, i \in \mathcal{F}\) 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) \[
\sum_{j \in \mathcal{R}} \beta_j = \beta_{\mathcal{F}}(\mathcal{R}).
\]

In the cooperative model, the repair process is divided into two rounds\(^3\). In the first round, each failed node downloads data from the helper nodes, and in the second round, the failed nodes download data from each other.

Definition 1.2 (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 \(f_{ij}(\{f_{ij}(C_j), j \in \mathcal{R}\})\) from each of other failed nodes \(C_{i'}, i' \in \mathcal{F}\backslash\{i\}\). We require that each failed node\(^1\) has access to all the downloaded information, 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 process of performing the repair. 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.

\(^1\)We note the use of the application-inspired term “download” for evaluating the functions \(f_{ij}\) and making their values available to the failed nodes. This term is used extensively in our paper.

\(^3\)In \[14\], the authors divide the repair process into \(T \geq 2\) rounds. The main reason for our choice of \(T = 2\) is that for all the cases considered in this paper, two rounds suffice to achieve the cut-set bound, and we conjecture that the same holds for the more general case. Apart from this, larger \(T\) leads to higher complexity of repair, and results in larger delay and also in unnecessarily complicated notation.
node \( C_i, i \in \mathcal{F} \) can be recovered from its own downloaded data \( f_{ij}(C_j), j \in \mathcal{R} \) and \( f_{iv}(\{f_{ij}(C_j), j \in \mathcal{R}\}), i' \in \mathcal{F}\{i\} \). The amount of downloaded data in this two-round repair process is
\[
\sum_{i \in \mathcal{F}} \left( \sum_{j \in \mathcal{R}} \dim_F (f_{ij}(C_j)) + \sum_{i' \in \mathcal{F}\{i\}} \dim_F (f_{iv}(\{f_{ij}(C_j), j \in \mathcal{R}\})) \right),
\]
where \( \dim_F(\cdot) \) is the dimension of the argument expressed as a vector over \( F \).

Given a code \( C \), define \( N_{ce}(C, \mathcal{F}, \mathcal{R}) \) and \( N_{co}(C, \mathcal{F}, \mathcal{R}) \) 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 1.3 (Repair bandwidth):** Let \( C \) be an \((n, k, l)\) MDS array code over a finite field \( F \). The \((h, d)\)-repair bandwidth of the code \( C \) under centralized/cooperative repair model is given by
\[
\beta_{ce}(h, d) := \max_{|\mathcal{F}|=h, |\mathcal{R}|=d, \mathcal{F} \cap \mathcal{R} = \emptyset} N_{ce}(C, \mathcal{F}, \mathcal{R}),
\]
\[
\beta_{co}(h, d) := \max_{|\mathcal{F}|=h, |\mathcal{R}|=d, \mathcal{F} \cap \mathcal{R} = \emptyset} N_{co}(C, \mathcal{F}, \mathcal{R}).
\] (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 1.1 (Cut-set bound [11, 8, 14]):** Let \( C \) be an \((n, k, l)\) MDS array code. For any two disjoint subsets \( \mathcal{F}, \mathcal{R} \subseteq [n] \) such that \(|\mathcal{F}| \leq r\) and \(|\mathcal{R}| \geq k\), we have the following inequalities:
\[
N_{ce}(C, \mathcal{F}, \mathcal{R}) \geq \frac{|\mathcal{F}||\mathcal{R}|}{|\mathcal{F}| + |\mathcal{R}| - k},
\] (2)
\[
N_{co}(C, \mathcal{F}, \mathcal{R}) \geq \frac{|\mathcal{F}|(|\mathcal{R}| + |\mathcal{F}| - 1)l}{|\mathcal{F}| + |\mathcal{R}| - k}.
\] (3)

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. If \( \beta_{ce}(h, d) \) (resp., \( \beta_{co}(h, d) \)) meets the bound (2) (resp., (3)) with equality, we say that the code \( C \) 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 \( C_j, j \in \mathcal{F}\{i\} \) 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 \( C_j, j \in \mathcal{R} \cup \mathcal{F}\{i\} \). 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 \( F \). 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. A self-contained rigorous proof of (3) is given in Appendix A as a part of the proof of Theorem 1.2.

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

**Theorem 1.2 (Cooperative model is stronger than centralized model):** Let \( C \) be an \((n, k, l)\) MDS array code and let \( \mathcal{F}, \mathcal{R} \subseteq [n] \) be two disjoint subsets such that \(|\mathcal{F}| \leq r\) and \(|\mathcal{R}| \geq k\). If
\[
N_{co}(C, \mathcal{F}, \mathcal{R}) = \frac{|\mathcal{F}|(|\mathcal{R}| + |\mathcal{F}| - 1)l}{|\mathcal{F}| + |\mathcal{R}| - k},
\] (4)
then
\[ N_{ce}(C, F, \mathcal{R}) = \frac{|F| |\mathcal{R}|}{|F| + |\mathcal{R}| - k}. \quad (5) \]

The proof of this theorem is given in Appendix A. The following arguments provide an intuitive explanation of its claim. 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 \( C_j, j \in \mathcal{R} \cup (\mathcal{F}\setminus\{i\}) \) 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 1.2, 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], [12], [17], MDS codes with the same property under the cooperative model are known only for some special values of \( h \) and \( d \). Specifically, the following is known. The authors of [14] constructed optimal codes for cooperative repair for the trivial case \( d = k \) and [15] presented a family of optimal codes for the repair of two erasures in the regime of low rate \( k/n \leq 1/2 \) (more precisely, [15] constructed explicit \((n, k)\) MDS codes with the \((2, d)\)-optimal repair property for any \( n, k, d \) such that \( 2k - 3 \leq d \leq n - 2 \)).

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 solve the case of repairing two erasures. More precisely, given any triple \((n, k, d)\) such that \( k \leq d \leq n - 2 \), we present an explicit \((n, k)\) MDS code with the \((2, d)\)-optimal repair property.
2) We solve the case of repairing multiple erasures from \( d = k + 1 \) helper nodes. More precisely, given any triple \((n, k, h)\) such that \( h \leq r - 1 \), we present an explicit \((n, k)\) MDS code with the \((h, k + 1)\)-optimal repair property.
3) We show that the any MDS code that affords cooperative optimal repair is also optimally repairable under the centralized model (see Theorem 1.2).

C. Organization of this paper

We start with the special case of \( h = 2 \) and \( d = k + 1 \) to illustrate the new ideas behind our constructions. These results are presented in Section II. Namely, in Section II-A we construct MDS codes \( C_{2,k+1}^{(0)} \) that can optimally repair the first two nodes (or any given pair of nodes) from any \( d = k + 1 \) helper nodes. In Section II-B we use this code as a “building block” to construct \((n, k)\) MDS codes \( C_{2,k+1} \) with the \((2, d)\)-optimal repair property. In Section III we deal with general values of \( d, k + 1 \leq d \leq n - 2 \). Similarly to the above, in Section III-A we construct a code \( C_{2,d}^{(0)} \) that supports optimal repair of the first two nodes, and in Section III-B we use it as a “building block” to construct MDS codes \( C_{2,d} \) with the \((2, d)\)-optimal repair property for general values of \( d, k + 1 \leq d \leq n - 2 \). In Section IV we construct \((n, k)\) MDS codes with \((h, d = k + 1)\)-optimal repair property for general values of \( h, 2 \leq h \leq r - 1 \). Again, in Section IV-A we handle the case of the first \( h \) failed nodes while in Section IV-B we extend the construction to any subset of \( h \) failed nodes. The corresponding codes are labelled as \( C_{h,k+1}^{(0)} \) and \( C_{h,k+1} \), respectively.

Note that, 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 III-B and IV-B reduce to the code of Section II upon appropriate adjustment of the parameters, such as taking $d = k + 1$ or $h = 2$, etc. The complete reduction scheme between the code families in this paper is as shown in Fig. 1.

![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 repair only of the first two (or the first $h$) erasures.]

D. Future directions

1) In [14], 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). Therefore the existence problem of MDS codes with $(h, d)$-optimal repair property for general values of $h$ and $d$ ($2 < h \leq r$ and $k + 1 < d \leq n - h$) remains an important open question.

2) Assuming that the existence of optimal codes for general values of $h$ and $d$ is established, the next question is whether two rounds of communication for repair suffice to achieve the cut-set bound (we conjecture that the answer is positive).

3) The repair problem of Reed-Solomon (RS) codes has attracted significant attention recently [7], [12], [18]–[23]. In particular, it is known that there exist RS codes with the $(h, d)$-optimal repair property under the centralized model [12]. Can this result be extended to the cooperative model (and are two rounds enough)?

4) Let us consider the regime where we fix $r$ and let $n$ grow. The sub-packetization value of our MDS code construction with the $(2, d)$-optimal repair property scales as $\exp(O(n^2))$ in this regime, which is much larger than its counterpart under the centralized model, where the sub-packetization value is $\exp(O(n))$ (see [11]). We conjecture that this is because the cooperative model is more restrictive than the centralized model. This raises an open question of resolving this conjecture, i.e., deriving a lower bound on sub-packetization for the cooperative model.

5) 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. 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 $k < n \leq |F| - 2$ 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 \( i = 1, 2, \ldots, n \), we write the \( i \)th 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 \( \lambda_{1,0}, \lambda_{1,1}, \lambda_{2,0}, \lambda_{2,1}, \lambda_{3}, \ldots, \lambda_{n} \) be \( n + 2 \) distinct elements of the field \( F \). The code \( \mathcal{C} \) is defined by the following 3 sets of parity check equations:

\[
\lambda_{1,0}^t c_{1,0} + \lambda_{2,0}^t c_{2,0} + \sum_{i=3}^{n} \lambda_{i}^t c_{i,0} = 0, \tag{6}
\]
\[
\lambda_{1,1}^t c_{1,1} + \lambda_{2,0}^t c_{2,1} + \sum_{i=3}^{n} \lambda_{i}^t c_{i,1} = 0, \tag{7}
\]
\[
\lambda_{1,0}^t c_{1,2} + \lambda_{2,1}^t c_{2,2} + \sum_{i=3}^{n} \lambda_{i}^t c_{i,2} = 0, \tag{8}
\]

where \( t = 0, 1, \ldots, r - 1 \). For each \( a = 0, 1, 2 \) the set of vectors \( (c_{1,a}, c_{2,a}, \ldots, c_{n,a}) \) 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 2.1:** For \( i = 1, \ldots, n \) 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 \( \mathcal{R} \subseteq \{3, 4, \ldots, n\} \), \( |\mathcal{R}| = k + 1 \), the values of \( c_{1,0}, c_{1,1}, \text{ and } \mu_{2,1} \) are uniquely determined by \( \{\mu_{i,1} : i \in \mathcal{R}\} \). Similarly, the values of \( c_{2,0}, c_{2,2}, \text{ and } \mu_{1,2} \) are uniquely determined by \( \{\mu_{i,2} : i \in \mathcal{R}\} \).

**Proof:** Let us prove the first statement. Adding (6) and (7), we obtain

\[
\lambda_{1,0}^t c_{1,0} + \lambda_{1,1}^t c_{1,1} + \lambda_{2,0}^t c_{2,0} + \sum_{i=3}^{n} \lambda_{i}^t c_{i,0} + \lambda_{2,0}^t c_{2,1} + \sum_{i=3}^{n} \lambda_{i}^t c_{i,1} = 0,
\]

\( t = 0, 1, \ldots, r - 1 \). Writing these \( r \) equations in matrix form, we obtain the following equality:

\[
\begin{bmatrix}
1 & 1 \\
\lambda_{1,0} & 1 \\
\lambda_{1,1} & 1 \\
\vdots & \vdots \\
\lambda_{1,0}^{r-1} & \lambda_{1,1}^{r-1}
\end{bmatrix}
\begin{bmatrix}
c_{1,0} \\
c_{1,1}
\end{bmatrix}
= 
\begin{bmatrix}
1 & 1 & 1 & \ldots & 1 \\
\lambda_{2,0} & \lambda_{3} & \lambda_{4} & \ldots & \lambda_{n} \\
\lambda_{2,0}^2 & \lambda_{3}^2 & \lambda_{4}^2 & \ldots & \lambda_{n}^2 \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
\lambda_{2,0}^{r-1} & \lambda_{3}^{r-1} & \lambda_{4}^{r-1} & \ldots & \lambda_{n}^{r-1}
\end{bmatrix}
\begin{bmatrix}
\mu_{2,1} \\
\mu_{3,1} \\
\mu_{4,1} \\
\vdots \\
\mu_{n,1}
\end{bmatrix}.
\tag{9}
\]

Note that \( r \geq 3 \) and define the polynomials \( p_0(x) := (x - \lambda_{1,0})(x - \lambda_{1,1}) \), and \( p_i(x) := x^i p_0(x) \) for \( i = 1, 2, \ldots, r - 3 \). Since \( \deg(p_i) < r \) for all \( i = 0, 1, 2, \ldots, r - 3 \), we can write

\[
p_i(x) = \sum_{j=0}^{r-1} p_{i,j} x^j.
\]

Define the \((r-2) \times r\) matrix

\[
P := 
\begin{bmatrix}
p_{0,0} & p_{0,1} & \ldots & p_{0,r-1} \\
p_{1,0} & p_{1,1} & \ldots & p_{1,r-1} \\
\vdots & \vdots & \ddots & \vdots \\
p_{r-3,0} & p_{r-3,1} & \ldots & p_{r-3,r-1}
\end{bmatrix}.
\]

\(^3\text{This proof draws on the ideas in } [4] \text{ Thm. IV.4].\)
Since for any $x$
\[
P[1 \ x \ x^2 \ \ldots \ \ x^{r-1}]^T = [p_0(x) \ p_1(x) \ p_2(x) \ \ldots \ \ p_{r-3}(x)]^T
= p_0(x)[1 \ x \ x^2 \ \ldots \ \ x^{r-3}]^T,
\]
we obtain
\[
\left[
\begin{array}{cccc}
1 & \lambda_{1,0} & \lambda_{1,1}^2 & \ldots & \lambda_{1,1}^{r-1} \\
1 & \lambda_{1,1} & \lambda_{1,1}^2 & \ldots & \lambda_{1,1}^{r-1}
\end{array}
\right]^T = 0,
\]
(10)
and
\[
P \left[
\begin{array}{cccc}
1 & 1 & 1 & \ldots & 1 \\
\lambda_{2,0} & \lambda_{3} & \lambda_{4} & \ldots & \lambda_{n} \\
\lambda_{2,0} & \lambda_{2,0}^2 & \lambda_{2,0}^3 & \ldots & \lambda_{2,0}^{n-1} \\
\vdots & \vdots & \vdots & \ldots & \vdots \\
\lambda_{2,0}^{r-1} & \lambda_{3}^{r-1} & \lambda_{4}^{r-1} & \ldots & \lambda_{n}^{r-1}
\end{array}
\right] = \left[
\begin{array}{cccccc}
p_0(\lambda_{2,0}) & p_0(\lambda_3) & p_0(\lambda_4) & \ldots & p_0(\lambda_n) \\
p_0(\lambda_{2,0}) & p_0(\lambda_3^2) & p_0(\lambda_4^2) & \ldots & p_0(\lambda_n^2) \\
p_0(\lambda_{2,0}) & p_0(\lambda_3^3) & p_0(\lambda_4^3) & \ldots & p_0(\lambda_n^3) \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
p_0(\lambda_{2,0}) & p_0(\lambda_3^{r-3}) & p_0(\lambda_4^{r-3}) & \ldots & p_0(\lambda_n^{r-3})
\end{array}
\right]
\]
(11)
Now let us multiply both sides of (9) on the left by $P$. On account of (11) and (10), we obtain the following equality:
\[
\left[
\begin{array}{cccc}
p_0(\lambda_{2,0}) & p_0(\lambda_3) & p_0(\lambda_4) & \ldots & p_0(\lambda_n) \\
p_0(\lambda_{2,0}) \lambda_{2,0} & p_0(\lambda_3) \lambda_{3} & p_0(\lambda_4) \lambda_{4} & \ldots & p_0(\lambda_n) \lambda_{n} \\
p_0(\lambda_{2,0}) \lambda_{2,0}^2 \lambda_{2,0} & p_0(\lambda_3) \lambda_{3}^2 \lambda_{3} & p_0(\lambda_4) \lambda_{4}^2 \lambda_{4} & \ldots & p_0(\lambda_n) \lambda_{n}^2 \lambda_{n} \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
p_0(\lambda_{2,0}) \lambda_{2,0}^{r-3} \lambda_{2,0}^{r-3} & p_0(\lambda_3) \lambda_{3}^{r-3} \lambda_{3}^{r-3} & p_0(\lambda_4) \lambda_{4}^{r-3} \lambda_{4}^{r-3} & \ldots & p_0(\lambda_n) \lambda_{n}^{r-3} \lambda_{n}^{r-3}
\end{array}
\right]
\left[
\begin{array}{c}
\mu_{2,1} \\
\mu_{3,1} \\
\mu_{4,1} \\
\vdots \\
\mu_{n,1}
\end{array}
\right] = 0
\]
(12)
Since $p_0(\lambda_{2,0}), p_0(\lambda_3), p_0(\lambda_4), \ldots, p_0(\lambda_n)$ are all nonzero, the vector $(\mu_{i,j}, i = 2, 3, \ldots, n)$ is a codeword in an $(n - 1, k + 1)$ generalized Reed-Solomon code. Therefore, for any $\mathcal{R} \subseteq \{3, 4, \ldots, n\}$, $|\mathcal{R}| = k + 1$, the values of $\mu_{2,1}, \mu_{3,1}, \ldots, \mu_{n,1}$ can be calculated from $\{\mu_{i,j} : i \in \mathcal{R}\}$. Once $\{\mu_{2,1}, \mu_{3,1}, \ldots, \mu_{n,1}\}$ are known, we can find $c_{1,0}$ and $c_{1,1}$ using (9). Thus we conclude that the values of $c_{1,0}, c_{1,1}$ and $\mu_{2,1}$ can be calculated from $\{\mu_{i,j} : i \in \mathcal{R}\}$. This completes the proof of the first statement in Lemma 2.1 and the proof of the second one is virtually identical.

This lemma implies that the first two nodes of $C$ 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 $\mu_{i,j}$ from the helper nodes $C_i, i \in \mathcal{R}$. According to Lemma 2.1, 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 $c_{2,0}, c_{2,1}$ 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 $C_1, i \in \mathcal{R} \cup \{1\}$. 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 $n \geq k + 3$ and a finite field $F, |F| \geq 2n$, we present an $(n, k, l = 3^m)$ MDS array code $C = C_{2, k + 1}$ over $F$, where $m = \binom{n}{2}$. When any two nodes of $C$ 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 $C$ by its parity-check equations, and we begin with some notation. Let $\{x_{ij}\}_{i \in [n], j \in \{0,1\}}$ be $2n$ distinct elements of the field $F$. Let $g$ be a bijection between the set of pairs $\{(i_1, i_2) : 1 \leq i_1 < i_2 \leq n\}$ and the set $\{1, 2, \ldots, m\}$. For concreteness, let
\[
g : (i_1, i_2) \mapsto \left(\frac{i_2 - 1}{2}\right) + i_1
\] (13)
\[(g)\text{ partitions the set } [m] \text{ into segments of length } (i_2 - 1), \text{ where } i_2 = 2, 3, \ldots, n\). Given an integer $a \in \{0, 1, \ldots, l - 1\}$, let $(a_m, a_{m-1}, \ldots, a_1)$ be the digits of its ternary expansion, i.e., $a = \sum_{j=0}^{m-1} a_j 3^j$. Define the following function
\[
f : [n] \times \{0, 1, \ldots, l - 1\} \rightarrow \{0, 1\}
\]
\[(i, a) \mapsto \left(\sum_{j=1}^{i-1} \mathbb{1}\{a_{g(j,i)} = 2\} + \sum_{j=i+1}^{n} \mathbb{1}\{a_{g(i,j)} = 1\}\right) \pmod{2},
\] (14)
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 does not include the digit labelled by $g(i_1, i_2)$ if and only if $i \not\in \{i_1, i_2\}$.

**Definition 2.1:** The code $C = C_{2,k+1}$ is defined by the following $rl$ parity check equations:
\[
\sum_{i=1}^{n} x_{i,f(i,a)} c_i, a = 0, \; t = 0, 1, \ldots, r - 1, \; a = 0, 1, \ldots, l - 1.
\]

For all $a = 0, 1, \ldots, l - 1$, the set of vectors $(c_1, a, c_2, a, \ldots, c_n, a)$ forms an $(n, k)$ MDS code, so $C$ is indeed an $(n, k, l)$ MDS array code.

Next we show that $C$ 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 $a = 0, 1, \ldots, l - 1$, $j \in [m]$, and $u = 0, 1, 2$, let
\[a(j, u) : = (a_m, \ldots, a_{j+1}, a, a_{j-1}, \ldots, a_1)\].

For $a = 0, 1, \ldots, l - 1$ 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 2.1, accounts for the $(2, k + 1)$ optimal repair property of the code $C$.

**Lemma 2.2:** Let $C_{i_1}$ and $C_{i_2}$, $i_1 < i_2$ be the failed nodes. For any set of helper nodes $\mathcal{R} \subseteq [n] \setminus \{i_1, i_2\}$, $|\mathcal{R}| = k + 1$ and any $a \in \{0, 1, \ldots, l - 1\}$, the values
\[c_{i_1, a(g_{12}, 0)}, c_{i_1, a(g_{12}, 1)}, \mu_{i_1}^{(a)}\]
are uniquely determined by the set of values $\{\mu_{i_1}^{(a)} : i \in \mathcal{R}\}$. Similarly, the values
\[c_{i_2, a(g_{12}, 0)}, c_{i_2, a(g_{12}, 2)}, \mu_{i_2}^{(a)}\]
are uniquely determined by the set of values $\{\mu_{i_2}^{(a)} : i \in \mathcal{R}\}$.

**Proof:** Recall that $a = 0, 1, \ldots, l - 1$ numbers the coordinates of the node, or the rows in the codeword array. For $u = 0, 1, 2$ the parity check equations corresponding to the row $a(g_{12}, u)$ are as follows:
\[
\sum_{i=1}^{n} x_{i,f(i,a(g_{12}, u))} c_i, a(g_{12}, u) = 0, \; t = 0, 1, 2, \ldots, r - 1.
\] (15)
According to definition of the function $f$ in (14) and the remarks made after it, we have

$$f(i, a(g_{i2}, 0)) = f(i, a(g_{i2}, 1)) = f(i, a(g_{i2}, 2)), \quad i \in [n] \setminus \{i_1, i_2\}$$

$$f(i_1, a(g_{i2}, 0)) = f(i_1, a(g_{i2}, 2)) \neq f(i_1, a(g_{i2}, 1)),$$

$$f(i_2, a(g_{i2}, 0)) = f(i_2, a(g_{i2}, 1)) \neq f(i_2, a(g_{i2}, 2)).$$

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

$$\lambda_i := \lambda_{i, f(i,a(g_{i2},0))} = \lambda_{i, f(i,a(g_{i2},1))} = \lambda_{i, f(i,a(g_{i2},2))}.$$  \hspace{1cm} (16)

Further, let

$$\lambda_{i,0}' := \lambda_{i, f(i,a(g_{i2},0))} = \lambda_{i, f(i,a(g_{i2},2))},$$

$$\lambda_{i,1}' := \lambda_{i, f(i,a(g_{i2},1))},$$

$$\lambda_{i,0}' := \lambda_{i, f(i,a(g_{i2},0))} = \lambda_{i, f(i,a(g_{i2},2))},$$

$$\lambda_{i,1}' := \lambda_{i, f(i,a(g_{i2},1))}.$$  \hspace{1cm} (17)

Notice that

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

$$\{\lambda_{i,0}', \lambda_{i,1}'\} = \{\lambda_{i,0}, \lambda_{i,1}\}$$

$$\{\lambda_{i,0}', \lambda_{i,1}'\} = \{\lambda_{i,0}, \lambda_{i,1}\}$$

$$\lambda_i \in \{\lambda_{i,0}, \lambda_{i,1}\}, \quad i \in [n] \setminus \{i_1, i_2\}.$$

Therefore $\lambda_{i,0}', \lambda_{i,1}', \lambda_{i,0}', \lambda_{i,1}'$, $\lambda_i$, $i \in [n] \setminus \{i_1, i_2\}$ are all distinct. Using the notation defined in (16)- (17), we can write (15) as

$$(\lambda_{i,0}') c_{i, a(g_{i2}, 0)} + (\lambda_{i,0}') c_{i, a(g_{i2}, 2)} + \sum_{i \in [n] \setminus \{i_1, i_2\}} \lambda_i c_{i, a(g_{i2}, 0)} = 0,$$

$$(\lambda_{i,1}') c_{i, a(g_{i2}, 1)} + (\lambda_{i,0}') c_{i, a(g_{i2}, 2)} + \sum_{i \in [n] \setminus \{i_1, i_2\}} \lambda_i c_{i, a(g_{i2}, 1)} = 0,$$

$$(\lambda_{i,0}') c_{i, a(g_{i2}, 2)} + (\lambda_{i,1}') c_{i, a(g_{i2}, 2)} + \sum_{i \in [n] \setminus \{i_1, i_2\}} \lambda_i c_{i, a(g_{i2}, 2)} = 0,$$

$$t = 0, 1, 2, \ldots, r - 1.$$

Now notice that up to a notational change, these equations have the same form as equations (6)-(8). Therefore, the remaining part of the proof follows from the same linear-algebraic transformations as the proof of Lemma 2.1 and there is no need to repeat them here.

This lemma implies that the nodes $C_{i_1}$ and $C_{i_2}$ can be repaired with optimal bandwidth. Namely, in the first round of the repair process, $C_{i_1}$ downloads the values in the set $\{\mu_{i,1}^{(a)} : a_{g_{i2}} = 0\}$ and $C_{i_2}$ downloads the values $\{\mu_{i,2}^{(a)} : a_{g_{i2}} = 0\}$ from each helper node $C_i$, $i \in \mathcal{R}$. This enables $C_{i_1}$ to find the values

$$\{c_{i_1,a} : a_{g_{i2}} = 0\} \cup \{c_{i_1,a(a_{g_{i2},1})} : a_{g_{i2}} = 0\} \cup \{\mu_{i,1}^{(a)} : a_{g_{i2}} = 0\}.$$

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

$$\{c_{i_2,a} : a_{g_{i2}} = 0\} \cup \{c_{i_2,a(a_{g_{i2},2})} : a_{g_{i2}} = 0\} \cup \{\mu_{i,2}^{(a)} : a_{g_{i2}} = 0\}.$$

In the second round, $C_{i_1}$ downloads $\{\mu_{i,2}^{(a)} : a_{g_{i2}} = 0\}$ from $C_{i_2}$, and $C_{i_2}$ downloads $\{\mu_{i,1}^{(a)} : a_{g_{i2}} = 0\}$ from $C_{i_1}$. After the second round, $C_{i_1}$ knows the values of all the elements in the set

$$\{c_{i_1,a(a_{g_{i2},u})} : a_{g_{i2}} = 0, u \in \{0, 1, 2\}\} = \{c_{i_1,a} : a \in \{0, 1, 2, \ldots, l - 1\}\},$$
and $C_{i_2}$ knows the values of all the elements in the set 
\[ \{c_{i_2,a(g_{i_2,u})} : a_{g_{i_2,u}} = 0, a \in \{0, 1, 2\}\} = \{c_{i_2,a} : a \in \{0, 1, 2, \ldots, l - 1\}\}, \]
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 $C_{i_1}, i \in R \cup \{i_2\}$, and $C_{i_2}$ downloads $l/3$ symbols of $F$ from each of the nodes $C_{i_2}, i \in R \cup \{i_1\}$. Therefore the total repair bandwidth is $2(k + 2)/3$, meeting the cut-set bound (3) with equality.

III. 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 $k + 1 \leq d \leq n - 2$, 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 $C = C^{(0)}_{2,d}$ over the field $F$ that has the following property. When the first two nodes of $C$ 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 $\lambda_{1,0}, \lambda_{1,1}, \ldots, \lambda_{1,s-1}, \lambda_{2,0}, \lambda_{2,1}, \ldots, \lambda_{3,0}, \lambda_{3,1}, \ldots, \lambda_{n}$ be $n - 2 + 2s$ distinct elements of the field $F$. Given an integer $a, 0 \leq a \leq s^2 - 2$, let $b_1(a), b_2(a)$ be the digits of its expansion to the base $s$:

\[ a = (b_2(a), b_1(a)). \]  

(18)

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

\[ \lambda_{1,b_1(a)}^t c_{1,a} + \lambda_{2,b_2(a)}^t c_{2,a} + \sum_{i=3}^{n} \lambda_i^t c_{i,a} = 0. \]  

(19)

Clearly, for a given $a$ the set of vectors $\{(c_{1,a}, c_{2,a}, \ldots, c_{n,a})\}$ forms an MDS code of length $n$ and dimension $k$. Therefore $C$ is indeed an $(n, k, s^2 - 1)$ MDS array code. Note that for $d = k + 1$, the code $C$ defined by (19) is the same as the code defined by (6)-(8) in Section II.

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

\[ \mu_{i,1}^{(v_2)} := \sum_{v_2=0}^{s-1} c_{i,s v_2 + v_1}, \quad v_2 \in \{0, 1, \ldots, s - 2\}; \]

\[ \mu_{i,2}^{(v_1)} := \sum_{v_2=0}^{s-1} c_{i,s v_2 + v_1}, \quad v_1 \in \{0, 1, \ldots, s - 2\}. \]

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

**Lemma 3.1:** Suppose that the failed nodes are $C_1, C_2$ and let $R \subseteq \{3, 4, \ldots, n\}, |R| = d$ be a set of $d$ helper nodes. For any $v_2 \in \{0, 1, \ldots, s - 2\}$, the values $\{c_{1,s v_2 + v_1}, v_1 = 0, 1, \ldots, s - 1\}$ and $\mu_{i,1}^{(v_2)}$ are uniquely determined by the set of values $\{\mu_{i,1}^{(v_2)} : i \in R\}$. Similarly, for any $v_1 \in \{0, 1, \ldots, s - 2\}$, the values $\{c_{2,s v_2 + v_1}, v_2 = 0, 1, \ldots, s - 1\}$ and $\mu_{i,2}^{(v_1)}$ are uniquely determined by the set of values $\{\mu_{i,2}^{(v_1)} : i \in R\}$.

**Proof:** The two parts of the claim are proved in the same way. We prove only the first of them. According to the definition (19), the parity-check equations that correspond to $a = s v_2, s v_2 + 1, \ldots, s v_2 + s - 1$ are as follows:

\[ \lambda_{1,v_1}^t c_{1,s v_2 + v_1} + \lambda_{2,v_2}^t c_{2,s v_2 + v_1} + \sum_{i=3}^{n} \lambda_i^t c_{i,s v_2 + v_1} = 0, \]
\[ t = 0, 1, \ldots, r - 1, \quad v_1 = 0, 1, \ldots, s - 1. \]

Summing these equations on \( v_1 \in \{0, 1, \ldots, s - 1\}, \) we obtain the equations

\[ \sum_{v_1=0}^{s-1} \lambda^t_{1,v_1} C_{1,sv2+v_1} + \lambda^t_{2,v_2} \mu_{2,1}^{(v_2)} + \sum_{i=3}^{n} \lambda^t_{i} \mu_{i,1}^{(v_2)} = 0, \quad t = 0, 1, \ldots, r - 1. \]

Let us write these \( r \) equations in matrix form. We obtain

\[
\begin{bmatrix}
1 & 1 & \ldots & 1 \\
\lambda_{1,0} & \lambda_{1,1} & \ldots & \lambda_{1,s-1} \\
\lambda_{2,0}^2 & \lambda_{2,1}^2 & \ldots & \lambda_{2,s-1}^2 \\
\vdots & \vdots & \ddots & \vdots \\
\lambda_{r,0}^{r-1} & \lambda_{r,1}^{r-1} & \ldots & \lambda_{r,s-1}^{r-1}
\end{bmatrix} \begin{bmatrix} C_{1,sv2} \\ C_{1,sv2+1} \\ \vdots \\ C_{1,sv2+s-1} \end{bmatrix} = - \begin{bmatrix} 1 & 1 & \ldots & 1 \\
\lambda_{2,v_2} & \lambda_{3} & \lambda_{4} & \ldots & \lambda_{n} \\
\lambda_{2,v_2}^2 & \lambda_{3}^2 & \lambda_{4}^2 & \ldots & \lambda_{n}^2 \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
\lambda_{r,v_2}^{r-1} & \lambda_{r-1}^{r-1} & \ldots & \lambda_{r,s-1}^{r-1}
\end{bmatrix} \begin{bmatrix} \mu_{2,1}^{(v_2)} \\ \mu_{3,1}^{(v_2)} \\ \mu_{4,1}^{(v_2)} \\ \vdots \end{bmatrix}.
\]

The proof will be complete once we show that the set of vectors \( \{\mu_{2,1}^{(v_2)}, \mu_{3,1}^{(v_2)}, \ldots, \mu_{n,1}^{(v_2)}\} \) forms a generalized Reed-Solomon code of length \( n - 1 \) and dimension \( d \). This can be done in the same way as in the proof of Lemma 2 above, and there is no need to repeat it here.

Let us show that this lemma implies that the first two nodes of \( C \) can be repaired with optimal bandwidth. In the first round, the first node \( C_1 \) downloads the values \( \{\mu_{i,1}^{(v_2)} : v_2 = 0, 1, \ldots, s - 2\} \) from each helper node \( C_i, i \in \mathcal{R}, \) and the second node \( C_2 \) downloads \( \{\mu_{i,2}^{(v_1)} : v_1 = 0, 1, \ldots, s - 2\} \) from each helper node \( C_i, i \in \mathcal{R}. \) From Lemma 3, we conclude that after the first round, \( C_1 \) knows the values

\[
c_{1,sv2+v_1}, \quad v_2 = 0, 1, \ldots, s - 2, \quad v_1 = 0, 1, \ldots, s - 1
\]

and \( \mu_{2,1}^{(v_2)}, \quad v_2 = 0, 1, \ldots, s - 2. \)

In the same way, \( C_2 \) knows the values

\[
c_{2,sv2+v_1}, \quad v_1 = 0, 1, \ldots, s - 2, \quad v_2 = 0, 1, \ldots, s - 1
\]

\[
\mu_{1,2}^{(v_1)}, \quad v_1 = 0, 1, \ldots, s - 2.
\]

In the second round, \( C_1 \) downloads the sums \( \mu_{1,2}^{(v_1)} : v_1 = 0, 1, \ldots, s - 2 \) from \( C_2, \) and \( C_2 \) downloads the sums \( \mu_{2,1}^{(v_2)}, v_2 = 0, 1, \ldots, s - 2 \) 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 \( k + 1 \leq d \leq n - 2, \) let \( s := d + 1 - k \) and let \( F \) be a finite field such that \( |F| \geq sn. \) We present an \((n, k, l = (s^2 - 1)^m)\) MDS array code \( C = C_{2,d} \) over the field \( F, \) where \( m := \binom{n}{2}. \) When any two nodes of \( C \) fail, the repair of each failed node can be accomplished by connecting to any \( d \) helper nodes and downloading \( (d + 1)!/((s + 1)!) \) 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 \( C \) by its parity-check equations, and we begin with some notation. Let \( \{\lambda_{ij}\}_{i \in [n], j \in \{0, 1, \ldots, s - 1\}} \) 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 \( \{1, 2, \ldots, m\} \) defined in (13). For every \( a = 0, 1, 2, \ldots, l - 1, \) we write its
expansion in the base \((s^2 - 1)\) as \(a = (a_m, a_{m-1}, \ldots, a_1)\), i.e., \(a = \sum_{j=0}^{m-1} a_{j+1}(s^2 - 1)^j\). Define the following function

\[
f : [n] \times \{0, 1, \ldots, l - 1\} \to \{0, 1, \ldots, s - 1\}
\]

\[
(i, a) \mapsto \left(\sum_{j=1}^{i-1} b_2(a_{g(j,i)}) + \sum_{j=i+1}^{n} b_2(a_{g(j,i)})\right) \pmod{s},
\]

where \(b_1(x)\) and \(b_2(x)\) form the digits of the expansion of \(x\) in the base \(s\); see definition (18). Note that when \(d = k + 1\), the function \(f\) defined in (20) is the same as the function defined in (14) in Section II-B.

**Definition 3.1:** The code \(\mathcal{C} = C_{2,d}\) is defined by the following \(rl\) parity check equations.

\[
\sum_{i=1}^{n} \lambda^t_{i,f(i,a)}c_{i,a} = 0, \quad t = 0, 1, 2, \ldots, r - 1, \quad a = 0, 1, 2, \ldots, l - 1.
\]

For a given \(a = 0, 1, \ldots, l - 1\) the set of vectors \(\{(c_{1,a}, c_{2,a}, \ldots, c_{n,a})\}\) 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 II-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 \(a = 0, 1, \ldots, l - 1\), \(j \in [m]\), and \(u \in \{0, 1, 2, \ldots, s^2 - 2\}\), let \(a(j, u) := (a_m, \ldots, a_{j+1}, u, a_{j-1}, \ldots, a_1)\). For \(a = 0, 1, \ldots, l - 1\) and \(i \in [n]\), we define

\[
\mu_{(a,v_2)}^{(a,v_2)} := \sum_{v_1=0}^{s-1} \lambda_{i,f(i,a)}c_{i,a}, \quad v_2 = 0, 1, \ldots, s - 2,
\]

\[
\mu_{(a,v_1)}^{(a,v_2)} := \sum_{v_2=0}^{s-1} \lambda_{i,f(i,a)}c_{i,a}, \quad v_1 = 0, 1, \ldots, s - 2,
\]

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

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

\[
\sum_{i=1}^{n} \lambda_{i,f(i,a(g_{12},u))}c_{i,a(g_{12},u)} = 0, \quad t = 0, 1, 2, \ldots, r - 1, \quad u = 0, 1, \ldots, s^2 - 2.
\]

According to definition of the function \(f\) in (20), 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

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

Again according to (20), for all \(u = 0, 1, 2, \ldots, s^2 - 2\), we have

\[
f(i_1, a(g_{12}, u)) = (f(i_1, a(g_{12}, 0)) + b_1(u)) \pmod{s},
\]

\[
f(i_2, a(g_{12}, u)) = (f(i_2, a(g_{12}, 0)) + b_2(u)) \pmod{s}.
\]

Therefore, we are justified in using the following notation:

\[
\lambda_i := \lambda_{i,f(i,a(g_{11},0))} = \lambda_{i,f(i,a(g_{11},1))} = \lambda_{i,f(i,a(g_{11},2))}, \quad i \notin \{i_1, i_2\}
\]
\[
\lambda'_{i,v} := \lambda_{i,v \oplus f(i_1,a(g_{i_1,0}))}, \quad \lambda'_{i_2,v} := \lambda_{i_2,v \oplus f(i_2,a(g_{i_2,0}))}, \quad v \in \{0, 1, \ldots, s - 1\}
\]

where \(\oplus\) is addition modulo \(s\). By (22), for every \(u = 0, 1, 2, \ldots, s^2 - 2\), we have
\[
\lambda_{i_1,f(i_1,a(g_{i_1,0}))} = \lambda_{i_1,b_1(u \oplus f(i_1,a(g_{i_1,0})))} = \lambda'_{i_1,b_1(u)}; \\
\lambda_{i_2,f(i_2,a(g_{i_2,0}))} = \lambda_{i_2,b_2(u \oplus f(i_2,a(g_{i_2,0})))} = \lambda'_{i_2,b_2(u)}.
\]

(24)

Notice that
\[
\{\lambda'_{i_0,0}, \lambda'_{i_1,1}, \ldots, \lambda'_{i,s-1}\} = \{\lambda_{i_0,0}, \lambda_{i_1,1}, \ldots, \lambda_{i,s-1}\} \text{ for } i \in \{i_1, i_2\},
\]

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

Therefore \(\lambda'_{i_1,0}, \lambda'_{i_1,1}, \ldots, \lambda'_{i_1,s-1}, \lambda'_{i_2,0}, \lambda'_{i_2,1}, \ldots, \lambda'_{i_2,s-1}, \lambda_i, \ldots, \lambda_i \in [n] \backslash \{i_1, i_2\}\) are all distinct. Using (23) and (24), we can write (21) as
\[
(\lambda'_{i_1,b_1(u)})^t c_{i_1,a(g_{i_1,0})} + (\lambda'_{i_2,b_2(u)})^t c_{i_2,a(g_{i_2,0})} + \sum_{i \in [n] \backslash \{i_1, i_2\}} \lambda_i^t c_{i,a(g_{i,0})} = 0
\]

\[
t = 0, 1, 2, \ldots, r - 1, \quad u = 0, 1, \ldots, s^2 - 2.
\]

These equations have exactly the same form as the equations in (19). Therefore the remainder of the proof of this lemma follows the steps in the proof of Lemma 3.1 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
\[
\bigcup_{v_2=0}^{s-2} \{\mu_{i_1,i_2}^{(a,v_2)} : a_{g_{i_1}} = 0\}
\]

(25)

from each helper node \(C_i, i \in \mathcal{R}\). In the same way, \(C_{i_2}\) downloads the set of elements
\[
\bigcup_{v_1=0}^{s-2} \{\mu_{i_2,i_1}^{(a,v_1)} : a_{g_{i_2}} = 0\}
\]

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 \(C_{i_1}, i \in \mathcal{R}\), i.e., the cardinality of the set in (25). Since each digit of \(a\) in its \((s^2 - 1)\)-ary expansion can take \(s^2 - 1\) possible values, \(|\{\mu_{i_1,i_2}^{(a,v_2)} : a_{g_{i_1}} = 0\}| = l/(s^2 - 1)\). The set in (25) is the union of \(s - 1\) such sets, so its cardinality is \((s - 1)l/(s^2 - 1) = l/(s + 1)\).

According to Lemma 3.2, after the first round, \(C_{i_1}\) knows the values of
\[
\left(\bigcup_{v_2=0}^{s-2} \bigcup_{v_1=0}^{s-1} \{c_{i_1,a(g_{i_1,0}) v_2+v_1} : a_{g_{i_1}} = 0\}\right) \cup \left(\bigcup_{v_2=0}^{s-2} \{\mu_{i_1,i_2}^{(a,v_2)} : a_{g_{i_1}} = 0\}\right),
\]

(26)

and \(C_{i_2}\) knows the values of
\[
\left(\bigcup_{v_1=0}^{s-2} \bigcup_{v_2=0}^{s-1} \{c_{i_2,a(g_{i_2,0}) v_2+v_1} : a_{g_{i_2}} = 0\}\right) \cup \left(\bigcup_{v_1=0}^{s-2} \{\mu_{i_1,i_2}^{(a,v_1)} : a_{g_{i_2}} = 0\}\right).
\]

(27)

In the second round of the repair process, the nodes \(C_{i_1}, C_{i_2}\) exchange the second terms in (26)-(27): namely, \(C_{i_1}\) downloads the elements in the set \(\bigcup_{v_1=0}^{s-2} \{\mu_{i_1,i_2}^{(a,v_1)} : a_{g_{i_2}} = 0\}\) from \(C_{i_2}\), and \(C_{i_2}\) downloads the elements in the set \(\bigcup_{v_2=0}^{s-2} \{\mu_{i_2,i_1}^{(a,v_2)} : a_{g_{i_2}} = 0\}\) from \(C_{i_1}\). After the second round, \(C_{i_1}\) knows the values of all the elements in the set
\[
\{c_{i_1,a(g_{i_1,0})} : a_{g_{i_1}} = 0, u \in \{0, 1, 2, \ldots, s^2 - 2\}\} = \{c_{i_1,a} : a \in \{0, 1, 2, \ldots, l - 1\}\},
\]

where \(\oplus\) is addition modulo \(s\).
and \( C_{i_2} \) knows the values of all the elements in the set
\[
\{ c_{i_2,a(g_{i_2,u})} : a_{g_{i_2,u}} = 0, u \in \{ 0, 1, 2, \ldots, s^2 - 2 \} \} = \{ c_{i_2,a} : a \in \{ 0, 1, 2, \ldots, l - 1 \} \},
\]
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 \( 1/(s + 1) \) symbols of \( F \) from each of the nodes \( C_i, i \in \mathcal{R} \cup \{ i_2 \} \), and \( C_{i_2} \) downloads \( 1/(s + 1) \) symbols of \( F \) from each of the nodes \( C_i, i \in \mathcal{R} \cup \{ i_2 \} \). Therefore the total repair bandwidth is \( 2(d + 1)l/(s + 1) \), meeting the cut-set bound \((3)\) with equality.

IV. Cooperative \((h, k + 1)\) optimal codes for any \( h \)

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 \( h = 2, \ldots, r - 1 \). More specifically, given any \( k < n \), any \( h \leq r - 1 \), and a finite field \( F \) of cardinality \( |F| \geq n + h \), we present an \((n, k, h + 1)\) MDS array code \( \mathcal{C} = \mathcal{C}^{(0)}_{h,k+1} \) 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 \( (\lambda_{ij}, i = 1, \ldots, h, j = 0, 1), \lambda_{h+1}, \lambda_{h+2}, \ldots, \lambda_n \) be \( n + h \) distinct elements of the field \( F \). The code \( \mathcal{C} \) is defined by the following parity check equations.

\[
\sum_{i=1}^{h} \lambda_{i,0} c_{i,0} + \sum_{i=h+1}^{n} \lambda_{i,0} c_{i,0} = 0, \quad t = 0, 1, \ldots, r - 1;
\]

\[
\lambda_{a,1} c_{a,a} + \sum_{i=h+1}^{n} \lambda_{i,0} c_{i,a} + \sum_{i=h+1}^{n} \lambda_{i,0} c_{i,a} = 0, \quad t = 0, 1, \ldots, r - 1, \quad a = 1, 2, \ldots, h.
\]

(28)

For every \( a = 0, 1, \ldots, h \), the set of vectors \( \{ (c_{1,a}, c_{2,a}, \ldots, c_{n,a}) \} \) 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 \([1]\).

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

The proof of this lemma is exactly the same as that of Lemma \([2,1]\) 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 \( \mu_{ij} \) from each helper node \( C_i, i \in \mathcal{R} \). According to Lemma \([4,1]\) after the first round, for every \( j \in [h] \), the node \( C_j \) knows the values of \( c_{j,0}, c_{j,j} \) and \( \{ \mu_{ij}, i \in [h] \setminus \{ j \} \} \). In the second round, every failed node \( C_{j,j} \in [h] \) downloads the sum \( \mu_{ij} \) from each of the other failed nodes \( C_i, i \in [h] \setminus \{ j \} \). 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 \( c_{j,0} + \mu_{ij}, i \in [h] \setminus \{ j \} \). 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 \( C_i, i \in \mathcal{R} \cup [h] \setminus \{ j \} \). 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 $h \leq r - 1$, and a finite field $F$ of cardinality $|F| \geq 2n$, we present an $(n,k,l = (h+1)^m)$ MDS array code $C = C_{h,k+1}$ over the field $F$, where $m := \binom{n}{h}$. The code $C$ has the property that for any $h$-subset $F$ of $[n]$, the repair of each failed node $C_i$, $i \in F$ 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 $C$ by its parity-check equations, and we begin with some notation. Let $\{\lambda_{ij}\}_{i,j \in [n], j \in [0,1]}$ be $2n$ distinct elements of the field $F$. Let $g$ be a bijection between the set of $h$-subsets $\{F : F \subseteq [n], |F| = h\}$ and the numbers $\{1,2,\ldots,m\}$. As in (13), the particular choice of $g$ does not matter; for instance, we can take

$$g(\{i_h, i_{h-1}, \ldots, i_1\}) = \sum_{j=0}^{h-1} \binom{i_h-j-1}{h-j} + 1 \text{ for all } n \geq i_h > i_{h-1} > \cdots > i_1 \geq 1,$$

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

$$f : [n] \times \{0,1,\ldots,l-1\} \rightarrow \{0,1\}$$

$$(i,a) \mapsto \left( \sum_{F \in \lambda_{i,a}, F \supseteq i} 1 \{a_{g(F)} = z(F,i)\} \right) \pmod{2},$$

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

**Definition 4.1:** The code $C = C_{h,k+1}$ is defined by the following $rl$ parity-check equations:

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

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

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

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

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

**Lemma 4.2:** Let $F = \{i_1,i_2,\ldots,i_h\}$ be the set of failed nodes. For any set of helper nodes $R \subseteq [n]\setminus F$, $|R| = k + 1$, any $j \in [h]$, and any $a \in \{0,1,\ldots,l-1\}$, the values of $c_{i_1,a}(g(F),0),c_{i_2,a}(g(F),j)$ and $\{\mu_{i,a}^{(j)} : i \in F\{i_j\}\}$ are uniquely determined by $\{\mu_{i,a}^{(j)} : i \in R\}$.

The proof of this lemma relies on the same ideas as the proofs of Lemmas 2.2 and 3.2. For completeness we outline it at the end of this section.

Let us explain why Lemma 4.2 implies that $C_{i_1,i} \in F$ can be repaired with optimal bandwidth. In the first round of the repair process, every failed node $C_{i_1,j} \in [h]$ downloads $\{\mu_{i_1,i}^{(j)} : a_{g(F)} = 0\}$ from each helper node $C_{i_1,i} \in R$. According to Lemma 4.2, after the first round, $C_{i_1}$ knows the values of

$$\{c_{i_1,a} : a_{g(F)} = 0\} \cup \{c_{i_1,a}(g(F),j) : a_{g(F)} = 0\} \cup \{c_{i_1,a} + c_{i_1,a}(g(F),j) : a_{g(F)} = 0, i \in F\{i_j\}\}.$$
In the second round of the repair process, every failed node $C_{ij}, j \in [h]$ downloads \( \{c_{ij,a} + c_{ij,a(g(F),j')} : a_{g(F)} = 0\} \) from each of the other failed nodes $C_{ij'}$, $j' \in [h]\setminus\{j\}$. As a result, $C_{ij}$ knows the values of all the elements in the set

\[
\{c_{ij,a(g(F),u)} : a_{g(F)} = 0, u = 0, 1, \ldots, h\} = \{c_{ij,a} : a \in \{0, 1, 2, \ldots, l - 1\}\},
\]

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

**Proof of Lemma 4.2** The parity-check equations that correspond to the rows labelled by $a(g(F),0)$, $a(g(F),1), \ldots, a(g(F),h)$ are as follows:

\[
\sum_{i=1}^{n} \lambda_{i,f(i,a(g(F),u))} c_{ij,a(g(F),u)} = 0, \quad t = 0, 1, 2, \ldots, r - 1, \quad u = 0, 1, 2, \ldots, h.
\]

(30) According to definition of the function $f$ in \(29\), if $i \notin F$, then the value of $f(i,a)$ does not depend on the digit of $a$ in position $g(F)$. Thus we have

\[
f(i,a(g(F),0)) = f(i,a(g(F),1)) = \cdots = f(i,a(g(F),h)), \quad i \in [n] \setminus F.
\]

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

\[
f(i_j,a(g(F),0)) \neq f(i_j,a(g(F),j)),
\]

\[
f(i_j,a(g(F),0)) = f(i_j,a(g(F),j')), \quad j' \in [h]\setminus\{j\}.
\]

Thus we are justified in using the following notation:

\[
\lambda_{i} := \lambda_{i,f(i,a(g(F),0))} = \lambda_{i,f(i,a(g(F),1))} = \cdots = \lambda_{i,f(i,a(g(F),h))}, \quad i \in [n] \setminus F;
\]

(31)

\[
\lambda'_{i,0} := \lambda_{i,f(i,a(g(F),0))} = \lambda_{i,f(i,a(g(F),j')), j \in [h], j' \in [h]\setminus\{j\}};
\]

(32)

\[
\lambda'_{i,1} := \lambda_{i,f(i,a(g(F),j'))}, \quad j \in [h].
\]

Notice that

\[
\lambda'_{i,0} \neq \lambda'_{i,1} \quad \text{and} \quad \{\lambda'_{i,0}, \lambda'_{i,1}\} = \{\lambda_{i,0}, \lambda_{i,1}\} \quad \text{for all} \quad j \in [h],
\]

\[
\lambda_i \in \{\lambda_{i,0}, \lambda_{i,1}\}, \quad i \in [n] \setminus F.
\]

Therefore the elements $\lambda'_{i,0}, \lambda'_{i,2}, \ldots, \lambda'_{i,n-1}, \lambda'_{i,1}, \lambda'_{i,2}, \ldots, \lambda'_{i,n-1}, \lambda_i, i \in [n] \setminus F$ are all distinct. Now we can write \(30\) as

\[
\sum_{j=1}^{h} (\lambda'_{i,j})^t c_{ij,a(g(F),u)} + \sum_{i \in [n] \setminus F} \lambda_{i}^t c_{ij,a(g(F),u)} = 0, \quad t = 0, 1, \ldots, r - 1;
\]

\[
(\lambda'_{i,n})^t c_{ij,a(g(F),u)} + \sum_{i \in [h]\setminus\{u\}} (\lambda'_{i,j})^t c_{ij,a(g(F),u)} + \sum_{i \in [n] \setminus F} \lambda_{i}^t c_{ij,a(g(F),u)} = 0
\]

\[
t = 0, 1, \ldots, r - 1; \quad u = 1, 2, \ldots, h.
\]

These equations have exactly the same form as the equations in \(23\). Therefore the remainder of the proof of Lemma 4.2 follows the steps in the proof of Lemma 4.1 (or Lemma 3.1), and we do not repeat them here.
Appendix A
Proof of Theorem 1.2

Let $C$ 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 $h := |F|$ and $d := |R|$. To shorten the expressions, below we use the following notation

$$D_i(R) = \sum_{j \in R} \dim_F(f_{ij}(C_j)), \quad D_i(F) = \sum_{i' \in F \setminus \{i\}} \dim_F(f_{i'j}(\{f_{i'j}(C_j), j \in R\}))$$

for the number of symbols of $F$ downloaded by $C_i \in F$ 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 $f_{ij}$ were introduced in Definition 1.2. 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 $F, R$. Our goal is to show that

$$\sum_{i \in F} (D_i(R) + D_i(F)) \geq \frac{h(h + d - 1)}{h + d - k} l. \quad (33)$$

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:

$$\{f_{ij}(C_j), i \in F, j \in R\}. \quad (34)$$

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 (34). Therefore, on account of the centralized version of the cut-set bound (2) we have

$$\sum_{i \in F} D_i(R) \geq \frac{hd}{h + d - k} l. \quad (35)$$

To bound the second term on the left-hand side of (33), 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 $C = (C_1, \ldots, C_n) \in C$ and (by a slight abuse of notation) use the same symbols $C_i, i = 1, \ldots, n$ for the associated random variables. For any $i \in [n]$ (in particular, for any $i \in F$) and any subset $S \subseteq R$ of the helper nodes of size $|S| = k - 1$, we have

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

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 $S \subseteq R, |S| = k - 1$ we have

$$H(C_i | f_{ij}(C_j), j \in S) = l \log_2 |F|. \quad (36)$$

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

$$H(C_i | f_{ij}(C_j), j \in R) \cup f_{i'j}(\{f_{i'j}(C_j), j \in R\}) : i' \in F \setminus \{i\} = 0. \quad (37)$$

Combining (36) and (37), and using Lemma A.1 below, we obtain that

$$H\left(\{f_{ij}(C_j), j \in R \setminus S\} \cup \{f_{i'j}(\{f_{i'j}(C_j), j \in R\}) : i' \in F \setminus \{i\}\}\right) \geq l \log_2 |F|. \quad (38)$$
Therefore, for any \( i \in \mathcal{F} \) and any \( S \subseteq \mathcal{R}, |S| = k - 1 \)
\[
\sum_{j \in \mathcal{R} \setminus S} \dim_F \left( f_{ij} (C_j) \right) + \sum_{i' \in \mathcal{F} \setminus \{i\}} \dim_F \left( f_{i'i} \left( \{f_{ij}(C_j), j \in \mathcal{R} \} \right) \right) \geq l \tag{39}
\]
(the left-hand side on the above line is the entropy of the left-hand side of \((38)\) under the uniform distribution on its arguments. Since the entropy is maximized for the uniform distribution, \((39)\) is implied by \((38)\). Note also the switching of the base of logarithms from 2 to \(|F|\).)

Let us sum \((39)\) over all subsets \( S \subseteq \mathcal{R} \) of size \(|S| = k - 1 \). Only the first term on the left-hand side depends on \( S \), and for every \( j \in \mathcal{R} \), the term \( \dim_F \left( f_{ij} (C_j) \right) \) appears for \( \binom{d-1}{k-1} \) different choices of \( S \). Thus we have
\[
\left( \frac{d}{k-1} \right) D_i(\mathcal{R}) + \left( \frac{d}{k-1} \right) D_i(\mathcal{F}) \geq \left( \frac{d}{k-1} \right) l, \quad i \in \mathcal{F}.
\]
Dividing both sides by \( \binom{d}{k-1} \), we obtain that for every \( i \in \mathcal{F} \),
\[
\frac{d-k+1}{d} D_i(\mathcal{R}) + D_i(\mathcal{F}) \geq l.
\]
Finally, let us sum the above inequalities on all \( i \in \mathcal{F} \). We obtain
\[
\frac{d-k+1}{d} \sum_{i \in \mathcal{F}} D_i(\mathcal{R}) + \sum_{i \in \mathcal{F}} D_i(\mathcal{F}) \geq hl. \tag{40}
\]
Multiplying \((35)\) on both sides by \( \frac{k-1}{d} \) and then adding it to \((40)\), we obtain the desired inequality \((33)\).

This completes the proof of \((3)\).

We are left to prove the claim that for a given code \( C \), \((4)\) implies \((5)\). Assuming \((4)\), we observe that there is a choice of the functions \( \{ f_{ij}, j \in \mathcal{R}; f_{i'i}, i' \in \mathcal{F} \setminus \{i\} : i \in \mathcal{F} \} \) such that \((33)\) holds with equality. This means that \((40)\) and all the inequalities preceding it in the proof, including \((35)\), hold with equality, but equality in \((35)\) means that \((5)\) holds true.

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

**Proof:** By the assumption we have \( H(XYZ) = H(YZ) \). Therefore,
\[
H(Z) \geq H(Z|Y) = H(YZ) - H(Y) = H(XYZ) - H(Y) \geq H(XY) - H(Y) = H(X|Y).
\]

**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 Symp. on the Foundations of Computer Science (FOCS), October 15-17, 2017, Berkeley, CA, pp. 216–227.

[8] 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.

[9] 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.

[10] 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.

[11] 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.

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

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

[14] K. W. Shum and Y. Hu, “Cooperative regenerating codes,” IEEE Transactions on Information Theory, vol. 59, no. 11, pp. 7229–7258, 2013.

[15] 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.

[16] 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.

[17] 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.

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

[19] 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.

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

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

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

[23] 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.