Distributed Data Storage with Minimum Storage Regenerating Codes - Exact and Functional Repair are Asymptotically Equally Efficient

Viveck R. Cadambe, Syed A. Jafar, Hamed Maleki
Electrical Engineering and Computer Science
University of California Irvine,
Irvine, California, 92697, USA
Email: \{vcadambe, syed, hmaleki\}@uci.edu

Abstract

We consider a set up where a file of size $M$ is stored in $n$ distributed storage nodes, using an $(n,k)$ minimum storage regenerating (MSR) code, i.e., a maximum distance separable (MDS) code that also allows efficient exact-repair of any failed node. The MDS property ensures that the original file can be reconstructed even if any $n-k$ storage nodes fail. When a node fails, a new node collects data from the remaining $n-1$ healthy nodes and repairs the failed node. The problem of interest in this paper is to minimize the repair bandwidth $B$ for exact regeneration of the failed node, i.e., the minimum data to be downloaded by the new node to replace the failed node by its exact replica. Previous work has shown that with random network coding, a bandwidth of $B = \frac{M}{k(n-k)}$ is necessary and sufficient for functional (not exact) regeneration, i.e., if the repaired new node need not be exactly identical to the failed node, but only information equivalent to it. It has also been shown using interference alignment based techniques that if $k \leq \max(n/2,3)$ then, surprisingly, there is no extra cost of exact regeneration over functional regeneration and the same repair bandwidth of $\frac{M(n-1)}{k(n-k)}$ suffices for exact regeneration. The practically relevant setting of low-redundancy, i.e., $k/n > 1/2$ remains open for $k > 3$ and it has been shown that there is an extra bandwidth cost for exact repair over functional repair in this case. In this work, we adopt into the distributed storage context an asymptotically optimal interference alignment scheme previously proposed by Cadambe and Jafar for large wireless interference networks. With this scheme we solve the problem of repair bandwidth minimization for $(n,k)$ exact-MSR codes for all $(n,k)$ values including the previously open case of $k > \max(n/2,3)$. Our main result is that, for any $(n,k)$, and sufficiently large file sizes, there is no extra cost of exact regeneration over functional regeneration in terms of the repair bandwidth per bit of regenerated data. More precisely, we show that $\lim_{M \to \infty} \frac{B}{M} = \frac{n-1}{k(n-k)}$. The result is analogous to the wireless interference channel setting where exact interference alignment through linear beamforming is seen to be infeasible for more than 3 users, but almost perfect alignment is achieved asymptotically by the Cadambe-Jafar scheme over a large number of signaling dimensions for any number of users.

I. INTRODUCTION

The problem of interest in this paper is to minimize the bandwidth required to exactly repair failed nodes in distributed storage systems. It is well known that maximum distance separable (MDS) codes can be used to reliably store data in distributed storage nodes. To see this, consider a scenario where a file of size $M$ is to be stored in $n$ distributed storage nodes. The file is split into $k$ equal parts of size $M/k$ and stored in the first $k$ storage nodes, also known as systematic nodes. The remaining $(n-k)$ nodes, known as parity nodes or non-systematic nodes, store data of the same size, i.e., $M/k$, adding redundancy to protect from failure of storage nodes. The parity nodes are designed so that a failure of up to $(n-k)$ storage nodes can be tolerated, i.e., the original file can be completely recovered from the data stored at any $k$ nodes out of the original $n$ nodes. Clearly, for this problem, storing the data using an $(n,k)$ MDS code suffices to achieve the required reconstruction criterion, since an MDS code protects the data from $(n-k)$ erasures. Now, consider the case where only 1 node fails, and a new node is introduced to replace the failed node. The total amount of data to be downloaded by the new node to regenerate a single failed node will be henceforth referred to as the repair bandwidth. Clearly, a repair bandwidth of $M$ suffices to repair a failed node since the new node can download data of total size $M$ from any $k$ of the remaining $n-1$
healthy nodes to reconstruct the failed node. However, note the inherent inefficiency in the solution - to reconstruct a node of size $M/k$, the newcomer downloads data of size $M$, i.e., $k$ times the size of the data to be repaired. A question of interest is whether this inefficiency is fundamental, or whether the node can be repaired with the new comer downloading data of size less than $M$. More specifically, the question of interest of this paper is what is the minimum repair bandwidth required to repair a failed node? The question of minimum repair bandwidth has been studied previously from two perspectives [1], [2], [3], [4], [5]. The first is called functional regeneration [1], [2] and the second is called exact (or systematic) regeneration [3], [4], [5].

The functional regeneration problem requires the new node to replace the failed node by a function of the data, so that the reconstructed new node, along with the other nodes satisfy the property of being an $(n,k)$ MDS code. In other words, the repaired node is information equivalent to the originally stored data. Note that in the functional regeneration problem, the data stored by the repaired node need not be identical to the data stored by the failed node; all that is required is that the repaired node along with the other nodes forms a MDS code. This problem has been shown to be equivalent to finding the capacity of a particular wired single-source multi-cast network. Since network coding achieves the cut-set bound in a single-source multi-cast network, the functional regeneration problem has been solved, and it is shown in [1] that the minimum bandwidth required is $B = \frac{M(n-1)}{k(n-k)}$. Note that since $\frac{M(n-1)}{k(n-k)}$ is smaller than $M$ (for $k < n$), the solution trivially implies that reconstruction of a single failed node requires a smaller repair bandwidth than reconstruction of $(n-k)$ failed nodes, by a factor of $\frac{(n-1)}{k(n-k)}$.

The focus of this paper is on the exact (or systematic) regeneration problem, where the new comer is required to replace the failed node by a replica, i.e., an identical copy of the failed node. Since exact regeneration ensures that a failed systematic node is replaced by a systematic node, the systematic structure of data storage is retained. There is a practical advantage of preservation of the systematic structure which ensures easy access to the data for a client, since the client can simply download it from the $k$ systematic nodes without any decoding. Note that the constraints for systematic or exact regeneration are stricter than the functional regeneration problem. Since any solution for the exact regeneration problem is also a solution to functional regeneration problem, $\frac{M(n-1)}{k(n-k)}$ serves as a lower bound to the minimum repair bandwidth for the exact repair problem. However, if the repair bandwidth of $\frac{M(n-1)}{k(n-k)}$ suffices, in general, has been an open question. It is this open question that is the focus of this paper.

A. Related Work and Summary of Contributions

The exact regeneration problem was formulated and solved for the special case of $n = 4, k = 2$ in [3]. The solution was further extended to the more general case of $k \leq \max(3, n/2)$ in [4], [5]. The results in all these cases yield the same surprising conclusion: there is no price for exact regeneration over functional regeneration, and a repair bandwidth of $\frac{M(n-1)}{k(n-k)}$ suffices even for exact regeneration. The solution for these cases stems from drawing parallels between the exact regeneration problem and the wireless interference channel [4]. Such parallels enable the use of the interference management technique of interference alignment [6], [7] for the exact regeneration problem. However, prior to this work, as far as we are aware, little was known about the minimum repair bandwidth for
\( k > \max(3, \frac{n}{2}) \). From a practical perspective, note that the previously unsolved case of \( k > \max(3, \frac{n}{2}) \) is important because this case corresponds to the amount of parity data (i.e., number of parity nodes) being smaller than the original file size (number of systematic nodes). This case is briefly studied in reference [5] where for \( k > \frac{n}{2} + 1 \), it is shown that the lower bound of \( \frac{M(n-1)}{k(n-k)} \) cannot be achieved using linear codes. The main contribution of this paper is to make progress in this open problem drawing inspiration from the interference alignment solution for the \( K \) user wireless interference channel in [8]. We argue that, while that lower bound on the repair bandwidth of \( \frac{M(n-1)}{k(n-k)} \) may not be sufficient in general as noted in [5], the repair bandwidth per bit of repaired data can indeed achieve this lower bound in the limit of large file sizes for any \( k < n \). More precisely, we show that

\[
\lim_{M \to \infty} \frac{B}{M} = \frac{n-1}{k(n-k)},
\]

i.e., for any \( k < n \), with sufficiently large amount of data, there is no cost of exact repair over functional repair in terms of repair bandwidth per bit of repaired data. An interesting insight of our solution is that the size of the symbol extension in wireless interference channels is analogous to the file size of our solution. Reference [8] shows that the optimal number of degrees of freedom of the interference channel cannot be achieved with finite symbol extensions (using linear schemes), but can only be achieved asymptotically in the limiting case of arbitrarily large symbol extensions. This is analogous to our result combined with that of reference [5]. The reference shows that the bound of \( \frac{M(n-1)}{k(n-k)} \) cannot be achieved exactly. Here, we complete the analogy between file size and symbol extensions in our main result by showing that while the bound is not exactly achievable, it is achievable asymptotically in the limit of large file sizes. We state our main result formally below.

**Theorem 1:** Consider any tuple \((n, k)\) such that \( n > k \). For a file of size \( M \) stored in \( n \) distributed storage nodes as a part of a \((n, k)\) MDS code, the minimum repair bandwidth \( B \) for exact regeneration of a (single) failed node satisfies

\[
\lim_{M \to \infty} \frac{B}{M} = \frac{n-1}{k(n-k)}.
\]

Equivalently, we can write

\[
B = \frac{M(n-1)}{k(n-k)} + o(M)
\]

Our approach to the problem is different from those of [5], [4] which use Cauchy matrices for code construction for \( k < \max(3, n/2) \). Before we proceed to prove the theorem, we take a closer look at a simple case of \((n = 4, k = 2)\), initially presented in [3], to describe the role of interference alignment in exact regeneration.

**II. The Role of Interference Alignment in Exact Regeneration : \( n = 4, k = 2 \)**

Consider the case where the \( n = 4, k = 2 \). Further, let the file to be stored be \([x_1 \ x_2]^T\) where \( x_i, i = 1, 2 \) are \( 2 \times 1 \) vectors over a finite field of size \( q \) denoted by \( \mathbb{F}_q \).

**Remark 1:** Note that for sufficiently large file sizes, the field size \( q \) is a design parameter. For the solution for \( n = 4, k = 2 \) presented here, any prime \( q \geq 3 \) suffices. Also by defining \( x_1, x_2 \) to be \( 2 \times 1 \) vectors, we are assuming that the file-size \( M = 4 \), i.e., 4 scalars over the field. For large file sizes, \( M \) can be treated as a design parameter, since a code for a specified \( M \) can be used for larger files by splitting the file into portions of size \( M \).

As in Figure 2 the first systematic node stores \( x_1 \) and the second systematic node stores \( x_2 \). Let the parity nodes \( m = 3, 4 \) store vectors of the form \( A_{m,1}x_1 + A_{m,2}x_2 \), where \( A_{m,i}, i = 1, 2, m = 3, 4 \) are \( 2 \times 2 \) matrices. Now, consider the case where \( A_{3,2} = A_{4,2} = I \) and

\[
A_{3,1} = \begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix} \quad A_{4,1} = \begin{bmatrix} 2 & 0 \\ 0 & 1 \end{bmatrix}.
\]

Note that \( x_1, x_2 \) can be reconstructed from any \( k = 2 \) of the \( n = 4 \) nodes. Now, consider the case where the first node fails. First, we present a naive solution which does not align interference. Note that the contents of the node, i.e. \( x_1 \) can be reconstructed by a new comer downloading 4 linear combinations (or equations) from any other two nodes. For example, it can be reconstructed by downloading the \( 2 \times 1 \) vector \( x_2 \) from node 2 and the \( 2 \times 1 \) vector \( A_{1,1}x_1 + A_{1,2}x_2 \) from node 3. In this case, note that among the 4 dimensions (corresponding to the 4 linear equations) at the new comer, two dimensions are occupied by the data to be reconstructed, i.e., the desired data \( x_1 \) and two dimensions are occupied by the undesired data or interference \( x_2 \).
However, a more efficient solution exists. By aligning the interference into 1 dimension, we can see that the node can be repaired by downloading only 3 linear combinations of the stored data from the remaining healthy nodes. To see this, consider the case where the new comer downloads a total of 3 linear combinations of the stored data, one from each remaining healthy node, as follows.

- \( v_2^T x_2 \) from node 2
- \( v_3^T (A_{3,1} x_1 + A_{3,2} x_2) \) from node 2
- \( v_4^T (A_{4,1} x_1 + A_{4,2} x_2) \) from node 3

where \( v_2 = v_3 = v_4 = [1 \ 1]^T \) (See Figure 2). As shown in Figure 2, since the interference aligns into 1 dimension at the new comer. Further, since

\[
\text{rank}( [A_{3,1}^T v_3 \ A_{4,1}^T v_4] ) = \text{dimension of } x_1 = 2,
\]

and \( [A_{3,1}^T v_3 \ A_{4,1}^T v_4] \) is linearly independent of the aligned interference \( v_2 \), the data storage node can reconstruct the desired two dimensional vector \( x_1 \) from the received 3 dimensional vector. The code can be shown to exactly repair any failed node with a repair bandwidth of 3, i.e., with the new comer collecting 3 linear combinations from the healthy nodes. We now proceed to extend this for general values of \((n, k)\) and prove our main result.

### III. Proof of Theorem 1

We begin by generalizing the setting described in the previous section. The total data is represented by the \( M/k \times k \) dimensional matrix \( [x_1 \ x_2 \ldots \ x_k] \), where \( x_i \) is an \( M/k \times 1 \) dimensional vector stored by systematic node \( i \in \{1, 2, \ldots, k\} \). Node \( j \), where \( j \in \{k+1, k+2, \ldots, n\} \) being a parity node stores the \( M/k \times 1 \) vector \( A_{j,1} x_1 + A_{j,2} x_2 + \ldots + A_{j,k} x_k \), where \( A_{j,i} \) is a \( M/k \times M/k \) square matrix for \( i \in \{1, 2, \ldots, k\} \). Henceforth, we assume that for \( j \leq k \),

\[
A_{j,i} = \begin{cases} 
0 & j \neq i \\
I & j = i 
\end{cases}, \forall i \in \{1, 2, \ldots, k\}.
\]

The above assumption implies that the data stored in node \( j \in \{1, 2, \ldots, n\} \) is the \( M/k \times 1 \) vector

\[
D_j = \sum_{i=1}^k A_{j,i} x_i.
\]
Note that $A_{j,i}$ for $j = k+1, k+2, \ldots, n$ are a design choice that define the code; these matrices will henceforth be referred to as the coding matrices. We need to choose these matrices so that the code is an MDS code, i.e., using any subset of $k$ nodes, the entire $M \times 1$ vector of data must be reconstructable. Thus, we need to ensure that
\[
\text{rank} \left( \begin{pmatrix}
  A_{j_1,1} & A_{j_1,2} & \cdots & A_{j_1,k} \\
  A_{j_2,1} & A_{j_2,2} & \cdots & A_{j_2,k} \\
  \vdots & \vdots & \ddots & \vdots \\
  A_{j_k,1} & A_{j_k,2} & \cdots & A_{j_k,k}
\end{pmatrix} \right) = M
\] (1)
for any distinct $j_1, j_2, \ldots, j_k \in \{1, 2, \ldots, n\}$.

Now, when a node fails, the new comer collects a $\beta \times 1$ vector from each of the remaining $(n - 1)$ healthy nodes where $\beta = \frac{B}{n-k}$, so that the total repair bandwidth is $B$. Our goal is to find the coding matrices $A_{j,i}$, $(j, i) \in \{k+1, k+2, \ldots, n\} \times \{1, 2, \ldots, k\}$ and design the $\beta \times 1$ vector to be downloaded by the new comer so as to meet the required bound (presented in the statement of the theorem). We now describe our solution assuming that a systematic node fails. We will later describe how the solution can be adapted to repair failures of parity nodes. Without any generality, let us assume that node 1 fails. We provide a linear solution to this problem, so that the $\beta \times 1$ vector downloaded by the new comer from node $j > 1$ to repair node 1 is $V_j^T D_j$, where $V_j$ is a $M/k \times \beta$ matrix. The matrices $V_j$ will be henceforth referred to as the repair vectors. The new comer now has to regenerate the $M/k \times 1$ vector $x_1$ using $(n - 1)$ vectors of the form $V_j^T D_j, j = 2, 3, \ldots, n - 1$, each of dimension $\beta \times 1$. Notice that the $(k - 1)$ vectors (of dimension $\beta \times 1$) downloaded using the $(k - 1)$ systematic nodes do not contain any information about the desired vector $x_1$ and can be interpreted as interference. Therefore, the new comer has, apart from the interference, $(n - k)$ vectors of dimension $\beta \times 1$ containing linear combinations of the desired data. Thus, the vectors available at the new comer can be described as follows.

- $(k - 1)$ vectors of the form $V_j^T x_j, 1 < j \leq k$ - these vectors are downloaded from the $(k - 1)$ healthy systematic nodes. They contain no information about the desired data, and will be used to cancel interference.

- $(n - k)$ vectors of the form $\sum_{i=1}^{k} A_{j,i} x_i, k < j \leq n$ - these vectors contain both the desired signal and components of the interference.

The goal of our solution will be to completely cancel the interference from the latter $(n - k)$ vectors using the former $(k - 1)$ vectors listed above, and then to regenerate $x_1$ using the latter $(n - k)$ vectors. In order to completely cancel the interference related to $x_i$ using $V_j^T x_i$ by linear techniques, we will need, $\forall j = k+1, k+2, \ldots, n$, and for some $\beta \times \beta$ matrix $\Lambda_j$,
\[
V_j^T A_{j,i} x_i = \Lambda_j V_j^T x_i
\]
\Rightarrow \text{rowspan}(V_j^T A_{j,i}) \subseteq \text{rowspan}(V_j^T),
\]
\Rightarrow \text{colspan}(A_{j,i} V_j) \subseteq \text{colspan}(V_i), i = 2, 3, \ldots, k
\] (4)
where (3) follows from the fact that the matrices $A_{j,i}$ and $V_i$ are picked independent of the data $x_i$, and therefore need to satisfy (2) for any data vector $x_j$.

While the above condition ensures that the entire interference can be cancelled, we also need to ensure that, on interference cancellation, the $(n - k)$ vectors of dimension $\beta \times 1$ are sufficient to reconstruct $x_1$. Note that after interference cancellation, each of the $(n - k)$ vectors is of the form $V_j^T A_{j,1} x_1$, for $j = k+1, k+2, \ldots, n$. For linear reconstruction, we need
\[
\Psi \begin{bmatrix}
  V_{k+1}^T A_{k+1,1} x_1 \\
  V_{k+2}^T A_{k+2,1} x_1 \\
  \vdots \\
  V_n^T A_{n,1} x_1
\end{bmatrix} = x_1
\]
\Rightarrow \Psi \begin{bmatrix}
  V_{k+1}^T A_{k+1,1} \\
  V_{k+2}^T A_{k+2,1} \\
  \vdots \\
  V_n^T A_{n,1}
\end{bmatrix} x_1 = x_1
for some $M/k \times (n-k)\beta$ matrix $\Psi$. Therefore, we need
\begin{equation}
\text{clospan}(\begin{bmatrix} A_{k+1,1}^T V_{k+1} & A_{k+2,1}^T V_{k+2} & \cdots & A_{n,1}^T V_n \end{bmatrix}) = \frac{M}{k}
\end{equation}

Therefore, our goal is to design $A_{j,i}$ and $V_l$ for $j \in \{k+1,k+2,\ldots,n\}$, $i \in \{1,2,\ldots,k\}$, $l = 2,3,\ldots,n$ so that

- The code is a $(n,k)$ MDS code.
- The interference is aligned appropriately so that it can be completely cancelled.
- The desired signal $x_1$ can be regenerated at the new comer.

Thus, essentially we need to pick $A_{j,i}$ and $V_l$ for $j \in \{k+1,k+2,\ldots,n\}$, $i \in \{1,2,\ldots,k\}$, $l = 2,3,\ldots,n$ so that \((1), (4), (5)\) are satisfied. Further, as noted in Remark 1, the field size $q$ and $M$ are also design choices (for large file sizes) that we can use to satisfy these conditions.

### A. The solution: Choosing $A_{j,i}, V_l, M$ and $q$

For $k \leq \max(3,n/2)$, the solutions of \([4], [5]\) design these matrices using Cauchy matrices to satisfy these conditions. Here, note that the conditions \((4), (5)\) are similar to the interference alignment conditions in the interference channel \([8]\). Specifically, \((4)\) is analogous to the condition that all the interference must align in the $K$-user interference channel, and \((5)\) is similar to the condition that the desired signal must be linearly independent for linear decoding in the interference channel \([8]\). These parallels with enable us to build a solution based on the asymptotically perfect interference alignment scheme of the same reference.

On noting that there are $\Gamma = (n-k)(k-1)$ alignment equations in \([4]\), like in \([8]\), we choose $M = k(n-k)\Delta^\Gamma$ and $\beta = (\Delta+1)^\Gamma$, where $\Delta \geq 1$ can be any integer.\footnote{The intuition for these choices of $M$ and $\beta$ will hopefully become clear later in this section for a reader unfamiliar with \([8]\).} For any value of $\Delta$, we show the existence of a field size $q$, matrices $A_{j,i}, i \in \{1,2,\ldots,k\}, j \in \{k+1,k+2,\ldots,n\}$ and $V_l, l \in \{2,3,\ldots,n\}$ so that \((1), (4), (5)\) are satisfied and the failed node can be repaired. Finally, we show that our code can be used to repair non-systematic nodes as well. Before we proceed to give a random coding based construction of the coding matrices and repair vectors, we will evaluate the repair bandwidth achieved by our scheme. Noting that our construction is applicable for any value of $\Delta$, we can make $\Delta$, a design parameter, arbitrarily large. As $\Delta \to \infty$, we have $M \to \infty$ and
\[
\lim_{M \to \infty} \frac{B}{M} = \lim_{\Delta \to \infty} \frac{(n-1)(\Delta+1)^\Gamma}{k(n-k)\Delta^\Gamma} = \frac{(n-1)}{k(n-k)}
\]

We now proceed to explain our construction of coding matrices and repair vectors satisfying the constraints of repair \((1), (4), (5)\). Our solution, unlike those in references \([4], [5]\), is a random coding solution. Specifically, we choose the coding matrices $A_{j,i}, i \in \{1,2,\ldots,k\}, j \in \{k+1,k+2,\ldots,n\}$ randomly. We then provide an expression for $V_l, l \in \{2,3,\ldots,n\}$ as a (random) function of $A_{j,i}$ so that \((4)\) is satisfied. Then we show for large field size $q$, that \((1)\) and \((5)\) are satisfied with a non-zero probability. This implies that there exists at least one choice of coding matrices $A_{j,i}$ so that all the desired conditions, i.e., \((1), (4), (5)\) are satisfied.

#### Design of Coding Matrices, $A_{j,i}$:
The alignment constraints, \((4)\), are similar to the alignment constraints for the interference channel (See equation (50) in \([8]\)). Note that the matrices $A_{j,i}$ play a role analogous to channel matrices in wireless interference channels \([8]\). Drawing inspiration from \([8]\), we choose the $M/k \times M/k$ dimensional matrices $A_{j,i} \forall j = k+1,k+2,\ldots,n$ to be random diagonal matrices with each diagonal entry of each matrix chosen independently and uniformly distributed over the non-zero elements of the field $F_q$. In other words, we choose
\[
A_{j,i} = \begin{bmatrix}
    a_{i,j}^1 & 0 & \cdots & 0 \\
    0 & a_{i,j}^2 & \cdots & 0 \\
    \vdots & \vdots & \ddots & \vdots \\
    0 & 0 & \cdots & a_{i,j}^{n}
\end{bmatrix}
\]

with all the diagonal entries chosen independent of each other and independent of all the diagonal entries of all other coding matrices, i.e., with $a_{i,j}^m$ chosen independent of $a_{i,j}^\tilde{m}$ from the non-zero elements of the field, for all $i \neq \tilde{i}$ or $j \neq \tilde{j}$ or $m \neq \tilde{m}$, where $i,\tilde{i} \in \{1,2,\ldots,k\}, j,\tilde{j} \in \{k+1,k+2,\ldots,n\}$ and $m,\tilde{m} \in \{1,2,\ldots,M\}$. Note that all the coding matrices are full rank since all the diagonal elements are non-zero. We later show that that this code is an MDS code with non-zero probability.
Design of Repair Vectors, $V_l$: Here, we provide a set of repair vectors that satisfy (4). We first set the columns of vectors $V_l$ (which are analogous to beamforming vectors in interference channels)

$$V_2 = V_3 = \ldots = V_k = V'$$

$$V_{k+1} = V_{k+2} = \ldots = V_n = V$$

where, $V$ and $V'$ are $M/k \times \beta$ dimensional matrices. Then the relations (4) can be re-written as

$$\text{colspan}(A_{j,i}V) \subseteq \text{colspan}(V'), i = 2, 3, \ldots, k$$

for $j = k+1, k+2, \ldots, n$. Note that there are $(k-1)(n-k) = \Gamma$ conditions contained in (7). We wish to find $V, V'$ so that all these conditions are satisfied.

Intuitive understanding of asymptotic alignment: Before we provide precise expressions for $V, V'$, we will intuitively explain the extent of alignment required to satisfy (4), (5). Since our bandwidth is restricted by $\beta$, we need rank$(V) \leq \beta = (\Delta + 1)^\Gamma$ and rank$(V') \leq (\Delta + 1)^\Gamma$. Further, noting that (5) implies $\sum_{j=k+1}^n \text{rank}(V_j) \geq \frac{M}{k}$, we get rank$(V) \geq \frac{M}{k(n-k)} = \Delta^\Gamma$. Therefore $V$ must have at least $\Delta^\Gamma$ non-zero linearly independent columns. In order to satisfy (7), the span of the $\Delta^\Gamma$ non-zero column vectors of the matrix

$$[A_{k+1,1}V A_{k+2,1}V \ldots A_{n,1}V A_{k+1,2}V A_{k+2,2}V \ldots A_{n,k}V]$$

should align in the space spanned by the $(\Delta + 1)^\Gamma$ column vectors of $V'$. For large values of $\Delta$, since $\frac{\Delta^\Gamma}{(\Delta+1)^\Gamma} \rightarrow 1$, and all the coding matrices have a full rank of $M/k$, we have $\frac{\text{rank}(A_{j,i}V)}{\text{rank}(V')} \rightarrow 1$ for any $j \in \{k+1, \ldots, n\}, i \in \{1, 2, \ldots, k\}$. From (7) this implies that $\text{colspan}(A_{j,i}V) \approx \text{colspan}(V')$. In other words, the alignment between the $\Gamma$ matrices on the left hand side of the $\Gamma$ relations indicated by (7) is asymptotically perfect for large $\Delta$. Next we return to the mathematical construction of the alignment scheme.

Following the arguments of [8], [9], we choose the set of non-zero column vectors of $V, V'$ as shown below

$$V = \left\{ \left( \prod_{j=k+1,\ldots,n} A_{j,i}^{\alpha_{j,i}} \right) w : \alpha_{k+1,2}, \ldots, \alpha_{n,k} \in \{0, 1, \ldots, \Delta - 1\} \right\}$$

$$V' = \left\{ \left( \prod_{j=k+1,\ldots,n} A_{j,i}^{\alpha_{j,i}} \right) w : \alpha_{k+1,2}, \ldots, \alpha_{n,k} \in \{0, 1, 2, \ldots, \Delta\} \right\}$$

where the entries of the $M/k \times 1$ column vector $w$ are chosen uniformly over the non-zero elements of the field and independent of all the coding matrices.

Thus, the elements of $V$ contain products of (diagonal) coding matrices corresponding to interference symbols contained in the parity nodes, with each matrix raised to an exponent that is allowed to take integer values from 0 upto $\Delta - 1$. Since there are $\Gamma = (k-1)(n-k)$ coding matrices and $\Delta$ possible distinct values for the exponent of each matrix, the total number of elements, i.e. column vectors, in $V$ is $\Delta^\Gamma$. Similarly, the total number of column vectors in $V'$ is $(\Delta + 1)^\Gamma$. To understand the notation better, consider, e.g., the case where $\Delta = 1$. Then, $V = w$, i.e., just one column vector, and $V'$ contains all the $2^\Gamma$ vectors of the form

$$A_{k+1,2}^{\alpha_{k+1,2}} A_{k+1,3}^{\alpha_{k+1,3}} \ldots A_{k+1,n}^{\alpha_{k+1,n}} A_{k+2,2}^{\alpha_{k+2,2}} \ldots A_{n,k}^{\alpha_{n,k}} w$$

where $\alpha_{j,i} \in \{0, 1\}$. For any general value of $\Delta$, the columns of $V$ are of the form

$$A_{k+1,2}^{\alpha_{k+1,2}} A_{k+1,3}^{\alpha_{k+1,3}} \ldots A_{k+1,n}^{\alpha_{k+1,n}} A_{k+2,2}^{\alpha_{k+2,2}} \ldots A_{n,k}^{\alpha_{n,k}} w$$

where $\alpha_{j,i} \in \{0, 1, \ldots, \Delta - 1\}$ and $V'$ has columns of the form

$$A_{k+1,2}^{\alpha_{k+1,2}} A_{k+1,3}^{\alpha_{k+1,3}} \ldots A_{k+1,n}^{\alpha_{k+1,n}} A_{k+2,2}^{\alpha_{k+2,2}} \ldots A_{n,k}^{\alpha_{n,k}} w$$

$^2$For convenience, we ignore the abuse in notation of these equations; the quantity on the left denotes the matrix, whereas the quantity on the right only denotes the set of non-zero columns of the matrix.
where $\alpha_{j,i} \in \{0, 1, \ldots, \Delta\}$. Note that the ordering of the matrices $A_{j,i}$ in the above notation is irrelevant, since the coding matrices, being diagonal, commute. This commuting property is the key to the alignment scheme. Because the ordering of matrices is irrelevant, it is readily verified that multiplying any column vector from $V$ by any of the $A_{j,i}$ involved, produces a column vector contained in $V'$. This is because multiplication by $A_{j,i}$ simply raises the corresponding exponent of the element in $V$ by one, but the elements of $V'$ already include all such terms. Since the set of columns of $A_{j,i}V$ is a sub-set of the columns of $V'$ for any $j \in \{k+1, k+2, \ldots, n\}, i \in \{2, 3, \ldots, k\}$, it is evident that this choice of repair vectors satisfies (7), and equivalently, (4).

**Proof of (7), (5)**: We have now chosen coding matrices and repair vectors so that the alignment constraints (4) are satisfied. We now need to show (1) and (5). In order to show that the matrices of (1) and (5) are full rank, it is enough to show that their determinants are non-zero. Notice that the determinant of the matrix of (1), i.e.,

$$
\begin{bmatrix}
A_{j_{1,1}} & A_{j_{1,2}} & \ldots & A_{j_{1,k}} \\
A_{j_{2,1}} & A_{j_{2,2}} & \ldots & A_{j_{2,k}} \\
\vdots & \vdots & \ddots & \vdots \\
A_{j_{k,1}} & A_{j_{k,2}} & \ldots & A_{j_{k,k}}
\end{bmatrix}
$$

is a polynomial in its entries. Note that there are $\binom{n}{k}$ polynomials of this kind, which can be represented, for $l = 1, 2, \ldots, \binom{n}{k}$, as

$$f_l : \mathcal{A} \rightarrow \mathbb{F}_q,$$

where

$$\mathcal{A} = \left\{a_{j,i}^m : j \in \{k+1, k+2, \ldots, n\}, i \in \{1, 2, \ldots, k\}, m \in \{1, 2, \ldots, (n-k)\Delta^\Gamma\}\right\} \rightarrow \mathbb{F}_q,$$

denotes all the diagonal entries of the coding matrices. In the appendix, we show that each of these polynomials is a non-zero polynomial.

Similarly, we need to show (5), i.e.,

$$\text{colspan}([A_{k+1,1}V \ A_{k+2,1}V \ \ldots \ A_{n,1}V]) = M_k^n,$$

where $V$ has $\Delta^\Gamma = \frac{M}{k(n-k)}$ non-zero columns chosen using (8). Using these non-zero columns (i.e., discounting the columns of $V$ which are zero) the above matrix is of dimension $\frac{M}{k} \times \frac{M}{k}$. Therefore, to show that this square matrix has a full rank of $\frac{M}{k}$, we need to show that its determinant is non-zero. Since a determinant is a polynomial function of its entries, the determinant expansion above is a polynomial

$$g : \mathcal{A} \cup \{w_1, w_2, \ldots, w_{(n-k)\Delta^\Gamma}\} \rightarrow \mathbb{F}_q,$$

where $w = [w_1 \ w_2 \ \ldots \ w_{(n-k)\Delta^\Gamma}]^T$. An argument very similar to Lemma 1 of [10] can be used to show that the polynomial formed by this matrix for our solution is a non-zero polynomial (See also Appendix III in [8]). Thus, the product $f_1(\cdot)f_2(\cdot)\ldots f_n(\cdot)$ is non-zero polynomial of $\mathcal{A} \cup \{w_1, w_2, \ldots, w_{(n-k)\Delta^\Gamma}\}$. Using Schwartz-Zippel Lemma, for large enough $q$, we have at least one choice of coding matrices and repair vectors such that these polynomials do not evaluate to non-zero, and therefore a solution exists so that (1), (5) are satisfied.

**Repair of Non-Systematic (Parity) nodes**: So far, we have discussed an achievable scheme for regenerating a systematic node. The code $A_{j,i}$ constructed here can also be used to regenerate a failed parity node in the same manner. To see this, suppose that a parity node, say node $k+1$, fails. The new comer intends to regenerate $D_{k+1}$. Let

$$x'_1 = D_{k+1} = \sum_{i=1}^{k} A_{k+1,i}x_i.$$

Since the code is an MDS code, using a change of basis, we can write

$$x_1 = A'_{1,1}x'_1 + \sum_{i=2}^{k} A'_{1,i}x_i,$$

$$x_j = A'_{j,1}x'_1 + \sum_{i=2}^{k} A'_{j,i}x_i, j = k+2, k+3, \ldots, n$$
where $A'_{j,i}$ are all diagonal. In other words, a change of basis can essentially transform the regeneration of a parity node to appear like regeneration of a systematic node, i.e., with nodes $2, 3, \ldots, k+1$ viewed as systematic nodes storing data $x_2, x_3, \ldots, x_k, x'_j$; nodes $1, k+2, k+3, \ldots, n$ are viewed as parity nodes using coding matrices $A^*_{j,i} \notin \{2, 3, \ldots, k+1\}$. Since all the coding matrices are diagonal, the problem can be solved in a similar manner as above, i.e., the vectors downloaded by the new comer can be constructed as in (9), (8) and be verified to satisfy a property similar to (4). The only thing that remains is to verify if a reconstruction criterion similar to (5) is satisfied. In order to show this, it is enough to show that all the (random) diagonal entries of any new coding matrix $A^*_{j,i}$ for $j \notin \{2, 3, \ldots, k+1\}$ are uniformly distributed over the non-zero entries of the field which are independent of each other and independent of all diagonal entries in the other new coding matrices, much like our original construction. Showing this independence property will ensure that our earlier proof of (5) is applicable. In order to show this independence property, we explicitly evaluate $A'_{j,i}$ for $j \notin \{2, 3, \ldots, k+1\}$ as follows.

$$
A'_{j,i} = \begin{cases} 
A^{-1}_{k+1,i} & j = 1, i = 1 \\
-A^{-1}_{k+1,i}A_{k+1,i} & j = 1, i \in \{2, 3, \ldots, k\} \\
A_{j,i}A^{-1}_{k+1,i} & j \in \{k+2, \ldots, n\}, i = 1 \\
A_{j,i} - A^{-1}_{k+1,i}A_{k+1,i} & j \in \{k+2, \ldots, n\}, i \in \{2, 3, \ldots, k\} 
\end{cases}
$$

(11)

Now, note that if $S_1$ and $S_2$ are two diagonal matrices with their diagonal entries drawn independently and uniformly distributed over the non-zero elements of the field, then each of the matrices $S_1S_2, S_1 + S_2, -S_2$ and $S_2^{-1}$ has diagonal entries uniformly distributed over the non-zero elements of the field. Further each of these matrices are independent of $S_1$. This implies that all the diagonal entries of $A'_{j,i}, j \notin \{2, 3, \ldots, k+1\}$ are distributed independently of each other, and uniformly distributed over the non-zero elements of $F_q$. Also, this property can be used in (11) to verify that all the entries of any coding matrix $A'_{j,i}$ are independent of the entries of all the entries of any other new coding matrix $A'_{j,i}$ for $i \neq j$ and $j \neq j'$. For example, $A'_{1,1}$ is independently distributed of $A_{1,2}'$ since the entries of $A_{k+1,2}$ are independent of $A_{k+1,1}'$ in our original code construction. Thus, the basis transformation preserves the required independence criteria and a property similar to (5) holds. This completes the proof.

IV. CONCLUSION

We have shown that, per bit of data to be reconstructed, surprisingly, there is no loss of exact regeneration over functional regeneration in terms of the amount of repair bandwidth per bit of repaired data, in the limit of large file sizes, regardless of the desired redundancy level. The result is in contrast with previous work in [5] where it is shown that there is an efficiency loss for exact regeneration over functional regeneration especially for low redundancy levels. However, note that the two results do not contradict each other. While our asymptotic alignment scheme can approach arbitrarily close to the cut-set bound on minimum repair bandwidth per bit of repaired data, the bound is not achieved with exact equality. Also unlike previous work in [4], [5] we do not provide explicit codes or specify the minimum field size, since our arguments are based on properties of random matrices. Directions for ongoing work include interference alignment solutions for exact repair for each point on the storage-bandwidth tradeoff curve.

APPENDIX

We intend to show that the determinant of the matrix in (11) is a non-zero polynomial in its entries. Assuming, without loss of generality, that $j_1, j_2, \ldots, j_k$ are in ascending order, let $j_1, j_2, \ldots, j_{k-m} \in \{1, 2, \ldots, k\}$ and $j_{k-m+1}, j_{k-m+2}, \ldots, j_k \in \{k+1, k+2, \ldots, n\}$. Therefore, we need to show that the determinant of the following
where the determinant can be shown to be equal to the determinant of the following matrix.

where each matrix is a non-zero polynomial of its entries.

P columns of to simplify P a matrix does not change its determinant except for its sign, we make the row and column exchange operations non-zero where P zero is equivalent to showing that the determinant of π where, therefore, we essentially need to show that the determinant formed by the above matrix is non-zero. Since the first \( m \) rows, we want the following matrix to be full rank.

We need to show that the determinant of the \( m \times m \) matrix \( P \) is non-zero. Note that we have

Therefore, we essentially need to show that the determinant formed by the above matrix is non-zero. Since the first \( (k - m) \frac{M}{k} \times (k - m) \frac{M}{k} \) matrix is the identity matrix, expanding the determinant along the first \( (k - m) \frac{M}{k} \) rows, the determinant can be shown to be equal to the determinant of the following matrix.

We need to show that the determinant of the \( m \frac{M}{k} \times m \frac{M}{k} \) matrix \( P \) is non-zero. Note that we have

where each \( a_{j,i} \) is independent of \( a_{j',i'} \) for \( i \neq i' \) or \( j \neq j' \) or \( l \neq l' \). Since interchanging the rows or columns of a matrix does not change its determinant except for its sign, we make the row and column exchange operations to simplify \( P \). Let the rows of \( P \) be \( r_1, r_2, \ldots, r_{mM/k} \). Now, we only need to show that the determinant of \( P' \) is non-zero where

where \( \pi_1 : \{1, 2, \ldots, mM/k\} \to \{1, 2, \ldots, mM/k\} \) is a permutation. Now, further, let \( c_1, c_2, \ldots, c_{mM/k} \) be the columns of \( P' \). We then perform column exchange operations of \( P' \) to get the matrix \( P'' = [c_{\pi_2(1)} \ c_{\pi_2(2)} \ \ldots \ c_{\pi_2(mM/k)}] \), where, \( \pi_2 : \{1, 2, \ldots, mM/k\} \to \{1, 2, \ldots, mM/k\} \) is also a permutation. Now, that the determinant of \( P \) is non-zero is equivalent to showing that the determinant of \( P'' \) is non-zero. Choosing the permutations \( \pi_1, \pi_2 \) as

where
it can be verified that the \( m^\frac{M}{k} \times m^\frac{M}{k} \) matrix \( P'' \) has a block diagonal structure, with \( \frac{M}{k} \) blocks of size \( m \times m \). The \( i \)th block of \( P'' \) is

\[
\begin{bmatrix}
    a_{k+1,(k-m)+1}^i & a_{k+1,(k-m)+2}^i & \cdots & a_{k+1,k}^i \\
    a_{k+2,(k-m)+1}^i & a_{k+2,(k-m)+2}^i & \cdots & a_{k+2,k}^i \\
    \vdots & \vdots & \ddots & \vdots \\
    a_{k+m,(k-m)+1}^i & a_{k+m,(k-m)+2}^i & \cdots & a_{k+m,k}^i
\end{bmatrix}
\]

Since the determinant of a block diagonal matrix is a product of the determinant of each of its blocks, and the determinant of the square matrix formed by the above block is a non-zero polynomial of its entries, the determinant of the matrix in (1) is a non-zero polynomial of its entries, as required.

REFERENCES

[1] A. Dimakis, P. Godfrey, M. Wainwright, and K. Ramchandran, “Network coding for distributed storage systems,” in IEEE INFOCOM, pp. 2000–2008, may 2007.

[2] Y. Wu, “Existence and construction of capacity-achieving network codes for distributed storage,” in IEEE International Symposium on Information Theory, pp. 1150–1154, 28 2009-july 3 2009.

[3] Y. Wu and A. Dimakis, “Reducing repair traffic for erasure coding-based storage via interference alignment,” in IEEE International Symposium on Information Theory, pp. 2276–2280, 28 2009-july 3 2009.

[4] C. Suh and K. Ramchandran, “Exact regeneration codes for distributed storage repair using interference alignment,” CoRR, vol. abs/1001.0107, 2010. http://arxiv.org/abs/1001.0107.

[5] N. B. Shah, R. K. V., P. V. Kumar, and K. Ramachandran, “Explicit codes minimizing repair bandwidth for distributed storage,” CoRR, vol. abs/0908.2984, 2009. http://arxiv.org/abs/0908.2984.

[6] S. Jafar and S. Shamai, “Degrees of freedom region for the MIMO X channel,” IEEE Trans. on Information Theory, vol. 54, pp. 151–170, Jan. 2008.

[7] M. Maddah-Ali, A. Motahari, and A. Khandani, “Communication over MIMO X channels: Interference alignment, decomposition, and performance analysis,” in IEEE Trans. on Information Theory, pp. 3457–3470, 2008.

[8] V. Cadambe and S. Jafar, “Interference alignment and the degrees of freedom of the K user interference channel,” IEEE Trans. on Information Theory, vol. 54, pp. 3425–3441, Aug. 2008.

[9] V. R. Cadambe and S. Jafar, “Reflections on interference alignment and the degrees of freedom of the K user interference channel,” IEEE Information Theory Society Newsletter, vol. 59, pp. 5–9, December 2009.

[10] Viveck R. Cadambe, Syed A. Jafar, “Interference Alignment and the Degrees of Freedom of Wireless X Networks”, IEEE Transactions on Information Theory, Vol. 55, No. 9, Sep. 2009, Pages: 3893-3908.