

A Network Coding Based Framework for Construction of Systematic Minimum Bandwidth Regenerating (MBR) Codes for Distributed Storage

Swanand Kadhe
Dept. of Electrical
Engineering,
Texas A&M University
kswanand1@tamu.edu

M Girish Chandra
Innovation Labs,
TATA Consultancy Services
Bangalore, India
m.gchandra@tcs.com

Balaji Janakiram
Idea Device Pvt. Ltd.,
Bangalore, India
b@ideadevice.com

ABSTRACT

Regenerating codes are a family of erasure correcting codes that are primarily designed to minimize the amount of data required to be downloaded to repair a failed node in a distributed storage system.

In this article, the construction of systematic Minimum Bandwidth Regenerating (MBR) codes based on random network coding, is presented. The repair model considered is the hybrid repair model, wherein, the source (message) symbols are exactly replicated, while the redundant (parity) symbols are replaced by their functionally equivalent symbols. It is showed that the random network coding based constructions can preserve the practically favorable systematic feature and still achieve the optimal trade off between storage and repair bandwidth, if the coding is performed by combining the judiciously selected source symbols. Unlike most of the schemes present in the literature, the proposed constructions do not pose any restriction on the number of nodes participating in repair or on the total number of nodes, and thus add reconfigurability to the system. Moreover, during the repair of systematic nodes, the proposed codes require less number of disk reads compared to most of the codes in the literature.

In the second half of the article, it is proven that the proposed constructions satisfy the necessary subspace properties of a linear exact regenerating code that are established in the literature. Further, rigorous analytical study of the effect of Galois field size on the probability of successful regeneration and reconstruction is carried out, and the results are validated using the numerical simulations.

1. INTRODUCTION

Large distributed storage systems often rely on multiple unreliable storage nodes, and reliability is achieved by introducing some form of redundancy. Though replication is the simplest way of achieving reliability, erasure coding

This work was carried out when Swanand Kadhe and Balaji Janakiram were with TCS Innovation Labs Bangalore.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

The 19th International Conference on Management of Data (COMAD), 19th-21st Dec, 2013 at Ahmedabad, India.

Copyright ©2013 Computer Society of India (CSI).

based techniques have been used as they minimize the storage overhead [1]. In particular, maximum distance separable (MDS) codes like Reed-Solomon (RS) codes are popular. In a system using an (n, k) RS code, a file is first divided into k blocks and then these are encoded to form n blocks, which are distributed across n nodes. To *reconstruct* the entire file, it is sufficient for a *data collector (DC)* to connect to any k out of n nodes.

However, a node storing a coded block may fail or leave the system, in which case it is necessary to *regenerate* the failed node using the existing nodes. In a system using an MDS code, it is required to reconstruct the entire file to recover a chunk of data stored on a failed node. This is clearly an inefficient way of regeneration, for the network bandwidth is often a critical resource. In their seminal work, [2], introduced the problem of efficiently recovering a failed node, coined as *repair problem*, and proposed the concept of *Regenerating Codes (RC)* as a solution.

In a system using an RC, a file of size B units is encoded and stored across n nodes, with each node capable of storing α units of data. When a node (say j th node) fails, a new node, called as *newcomer*, downloads β units of data from any d of the remaining nodes to *regenerate* the failed node (as shown in Fig. 1). The $d\beta$ units of data required for the regeneration is called as the *repair bandwidth*. The objective of the regenerating codes is to reduce this repair bandwidth. In addition, at any point of time, the system should maintain MDS-code property, i.e., connecting to any k out of n nodes should be sufficient to *reconstruct* the entire original file (see Fig. 2).

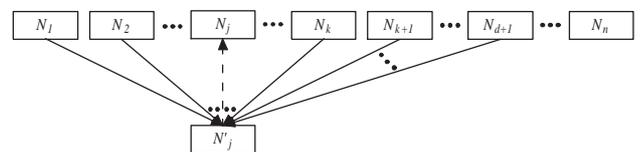


Figure 1: Schematic showing regeneration process for an (n, k, d) Regenerating code, with each node capable of storing α units of data and the amount of data transferred by each node to the newcomer N'_j is β .

It has been shown that the evolution of information flow

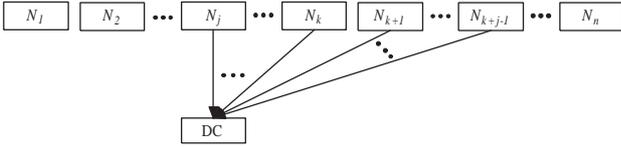


Figure 2: Schematic showing reconstruction process for an (n, k, d) Regenerating code, with each node capable of storing α units of data and the amount of data transferred by each node to the DC is β .

as nodes leave and join the storage network represents the information flow graph. The cutset analysis of the information flow graph has been utilized to establish that there exists a trade-off between storage per node (α) and repair bandwidth ($d\beta$) [3]. The codes on one extremal point of the optimal storage-repair bandwidth trade-off curve that minimize the repair bandwidth are called as Minimum Bandwidth Regenerating (MBR) codes; whereas, the codes on the other extremal point that minimize storage per node are called as Minimum Storage Regenerating (MSR) codes.

The regeneration of a failed node can be carried out according to one of the three repair models: functional repair, exact repair, or exact repair of systematic data (also known as hybrid model). Functional repair model follows relaxed notion of repair - the regenerated node need not be same as the failed one, but combined with the existing nodes the system continues to satisfy the MDS-code property (that is any k out of n nodes are sufficient to reconstruct the entire original data). In case of exact repair, the regenerated node is required to be an exact replica of the failed one. The third repair model, exact repair of systematic data, presumes that the code is systematic (i.e., coded symbols contain original data symbols in uncoded form). The nodes containing the systematic part are termed as systematic nodes, while other nodes are termed as non-systematic (or parity) nodes. The exact repair of systematic data is a hybrid model which combines exact and functional models by following exact repair for regenerating the systematic nodes and functional model for repairing the non-systematic nodes. It is important to emphasize that the systematic feature is practically desirable, since it allows a DC to connect to the systematic nodes and directly download the data without requiring any decoding.

The repair problem under the functional repair model has been completely characterized by using the cut-set analysis of the information flow graph associated with the storage network. Moreover, considerable amount of research has been carried out on designing the codes for exact repair model in recent years. An excellent survey of various code constructions has been carried out in [4]. To the best of our knowledge, the best known constructions in the literature can be summarized as follows (see [4] for details). A product-matrix based code construction was proposed in [5]. The codes match the cut-set bound for all $n, k, d \in [k, n-1]$ for MBR point; and for all $n, k, d \in [2k-2, n-1]$ for MSR point. Reference [6] proposed common eigenvector based framework for MSR point for $k/n \leq 1/2$ and $d \in [2k-1, n-1]$. This construction is in fact a generalization of [7], which focussed on repair of only systematic nodes. Hybrid model

has been considered for MSR point in [8] for $d = k+1$. On the one hand, the recent research has produced the code constructions that achieve the cut-set bound on the repair bandwidth for almost all the values of parameters for which the cut-set bound is known to be achievable. On the other hand, several other issues that affect the repair in a distributed storage system have started getting attention; e.g., reducing the amount of data to be read from a storage node during the repair (termed as number of disk reads), analyzing the influence of network topology etc. (see [4]).

We observe that the hybrid repair model (exact repair of systematic data) has received considerably less attention than the exact repair model, even though it has the potential to facilitate many favorable practical aspects: (a) it allows the coding scheme to be systematic which can facilitate faster download; (b) it provides reconfigurability to the system through dynamic and less constrained functional regeneration of parity blocks.

Motivated by these features, we seek to find out whether it is possible to construct the systematic regenerating codes using random network coding schemes following the hybrid repair model. Specifically, we present a coding scheme for MBR point which is optimal for all $n, k, d \in [k, n-1]$. The limitation of the proposed coding scheme is that it requires all the surviving systematic nodes to participate during the recovery of a failed node.¹

It is important to mention that that, even though network coding based regenerating codes are already present in the literature [2, 3, 9], these codes are inherently non-systematic. They linearly combine all the data blocks (either randomly or deterministically) for encoding, and follow functional model for regeneration. In this article, we show that, instead of combining all the data blocks, if the data blocks to be combined for each coded block are judiciously selected, then it is possible to achieve systematic feature while using random network coding. Furthermore, this selective combining reduces the number of disk reads during the recovery of systematic nodes.

In the second half of the paper, we focus on the analysis of required Galois field size for the proposed coding scheme. Most of the network coding based constructions in the literature rely on the high field size, but does not comment on the effect of field size on the performance. However, for practical implementation, it is important to understand how the field size affects the performance. Therefore, we present the rigorous analysis of Galois field size on the recovery performance of the proposed MBR codes and corroborate the analysis with numerical simulations.

In summary, the main contributions of this work can be stated as follows.

- We propose a random network coding based framework for construction of systematic MBR codes which match the cut-set bound for all feasible values of parameters n, k, d .
- We show that the proposed codes satisfy the subspace properties given in [10, 7], which are necessary for a linear exact regenerating code.
- We rigorously analyze the effect of Galois field size on the probabilities of successful regeneration and reconstruction for the proposed coding scheme.

¹It should be noted that this requirement is also present in many other coding schemes [7, 6].

In addition, we compare our framework with explicit exact repair constructions of [5]. We find out that our constructions continue to possess some of the key features present in the product-matrix framework [5]; for example, our construction spans entire range of parameters (n, k, d) as in case of [5]. In fact, we note that our constructions can be advantageous in certain scenarios. In particular, we would see that the proposed codes require less amount of data to be read while regenerating a failed systematic node compared to product-matrix framework. Also, proposed codes do not require rearranging the downloaded data when a DC connects to systematic nodes as required in product-matrix constructions. Moreover, we point out some observations on how the proposed framework can be extended to Minimum Storage Regenerating (MSR) point.

At this juncture, we would like to mention our previous work [11], which presents a low-complexity, practically amenable MBR code construction. The construction presented in [11], though sub-optimal, fits in the framework discussed in this article and can be considered as a precursor to the present work.

The rest of the article is organized as follows. In section 2, we describe the system model. The code construction for MBR point is presented in section 3. The study of subspace properties and the analysis of the effect of field size on the performance is presented section 4. Extending the proposed framework to MSR point is briefly touched upon in section 5. Finally, section 6 concludes the article.

2. SYSTEM MODEL

Consider a file consisting of B blocks to be stored across n nodes after coding, with each node capable of storing α blocks of data. The distributed system should satisfy *MDS property* - any k out of n nodes should be sufficient to reconstruct the entire file. When a node fails, it is regenerated according to hybrid model by downloading β blocks each from d ($k \leq d \leq n - 1$) surviving nodes.

For the sake of simplicity, we assume that each data block consists of only one symbol. Then, let $\mathcal{X} = \{x_1, x_2, \dots, x_B\}$ denote the set containing all the source symbols. Each source symbol $x_i \in \mathbb{F}_q$, where \mathbb{F}_q denotes Galois field of size q . Let $y_{(i,j)}$ denote the i th symbol stored in the j th storage node. The code is in systematic form, i.e., coded symbols contain original data symbols in uncoded form. We assume that first k nodes contain the source symbols and they are called as systematic nodes; remaining nodes are called as non-systematic or parity nodes. It is assumed that, in regeneration of a failed node, all the active systematic nodes always participate. Notice that this assumption has also been considered in [7, 6]. Let \mathcal{D}_j denote the subset of nodes participating in the regeneration of j th node.

To obtain the non-systematic (parity) symbols, random linear network coding (RLNC) [12] is used. Under conventional RLNC, non-systematic symbols are formed by linearly combining the source symbols with randomly chosen coefficients. Mathematically, if $y_{(i,j)}$ is a non-systematic symbol, we have

$$y_{(i,j)} = \sum_{l=1}^B c_{(i,j)}^l x_l, \quad 1 \leq i \leq \alpha, k+1 \leq j \leq n, \quad (1)$$

where $c_{(i,j)}^l$, $1 \leq l \leq B$, are coding coefficients chosen uniformly at random and independently over \mathbb{F}_q ; and the addi-

tion is performed over \mathbb{F}_q . Each node stores αB code coefficients corresponding to α symbols stored on it.² When newcomer or DC connects to a storage node, it also downloads corresponding code coefficients along with the appropriate symbols.

Instead of randomly combining all the source symbols to obtain a coded symbol, we propose to perform the combinations selectively. In particular, we divide the source symbols into different subsets $\mathcal{X}_i \subset \mathcal{X}$. Then, a coded symbol $y_{(i,j)}$ is obtained by randomly combining the elements of some subset \mathcal{X}_m , i.e.,

$$y_{(i,j)} = \sum_{x_l \in \mathcal{X}_m} c_{(i,j)}^l x_l, \quad 1 \leq i \leq \alpha, k+1 \leq j \leq n, \quad (2)$$

where coefficients $c_{(i,j)}^l$ are chosen uniformly at random and independently over \mathbb{F}_q ; and the addition is performed over \mathbb{F}_q . Therefore, in this case, a coded symbol is a linear functional of only the elements of \mathcal{X}_m ; this is denoted by $y_{(i,j)} = \mathcal{L}_{(i,j)}\{\mathcal{X}_m\}$. The specific construction of subsets is discussed in the next section.

3. MBR CODE CONSTRUCTION

The optimal α and β for MBR point can be shown as [3]:

$$(\alpha_{MBR}, \beta_{MBR}) = \left(\frac{2Bd}{2kd - k^2 + k}, \frac{2B}{2kd - k^2 + k} \right) \quad (3)$$

Thus, we have

$$B = \frac{\beta k(2d - k + 1)}{2} \quad (4)$$

and

$$\alpha = \beta d. \quad (5)$$

From (4) and (5), it can be seen that $\alpha \geq \frac{B}{k}$. Thus, first k nodes need to store $\frac{\beta(k-1)}{2}$ parity symbols along with the $\frac{\beta(2d-k+1)}{2}$ systematic symbols. We call these parity symbols stored on systematic nodes as *additional symbols*, for systematic nodes need to carry these extra symbols in addition to the source symbols.

The code constructions presented are for $\beta = 1$, similar to the most of the coding schemes in the literature. For higher values of β , codes can be obtained by concatenating the codes for $\beta = 1$ (see [5]). Note that, if k is even, half the systematic nodes store $\frac{(k-1)}{2} - 1$ additional symbols, whereas the remaining half nodes store $\frac{(k-1)}{2} + 1$ additional symbols. We call this case as unequal additional symbols case. In case of odd k , all the systematic nodes store the same number of additional symbols and is called as equal additional symbols case. We start with specific examples of the proposed coding scheme to clarify the structure and then present the generic code construction as a function of parameters k, n, d, α , and β .

3.1 Example for MBR code

Let us consider equal additional symbols case first. Let $k = 3, n = 6, d = 4$, and $\beta = 1$. This gives $B = 9$ and $\alpha = 4$. The set \mathcal{X} containing all source symbols is divided into $k = 3$

²Corresponding to systematic symbols, we assume that the columns of $B \times B$ identity matrix are stored as coding coefficients.

subsets. Thus, $\mathcal{X}_1 = \{x_1, x_4, x_7\}$, $\mathcal{X}_2 = \{x_2, x_5, x_8\}$, and $\mathcal{X}_3 = \{x_3, x_6, x_9\}$. The symbols in subset \mathcal{X}_j are stored in the j th systematic node. The code construction is shown in the following equation in the form of a matrix $\mathbf{Y}_{(n,k,d)}$. The j th column of the matrix represents the symbols to be stored on node j . As stated earlier, $y_{(i,j)} = \mathcal{L}_{(i,j)} \{\mathcal{X}_m\}$ denotes the coded symbol formed by the random linear combination of the symbols in the set \mathcal{X}_m .

Notice that the additional symbol stored in a systematic node is a random linear combination of the source symbols stored in the cyclically next (or previous) systematic node. For the non-systematic node j , the i th parity symbol ($1 \leq i \leq k$) is obtained by randomly combining the source symbols in set \mathcal{X}_i , i.e., $y_{(i,j)} = \mathcal{L}_{(i,j)} \{\mathcal{X}_i\}$. The remaining parity symbols are random combinations of all the source symbols ($y_{(i,j)} = \mathcal{L}_{(i,j)} \{\mathcal{X}\}$). We call these symbols as *globally combined parity symbols*. It is assumed that all the coefficient vectors corresponding to the linear combinations are linearly independent of each other. This can be achieved by selecting the arbitrarily high field size [12].

Let us consider the regeneration process assuming that the first node is failed. Also, assume that the newcomer downloads $\beta = 1$ symbol each from nodes 2, 3, 5, and 6. Note that surviving systematic nodes should participate in the regeneration process. Now, nodes 3, 5, and 6 give the linear combinations in \mathcal{X}_1 along with the corresponding coefficient vectors, forming three linear equations in three elements of \mathcal{X}_1 . Thus, the newcomer can decode the symbols in \mathcal{X}_1 , under the assumption of linear independence due to high field size. Node 2 computes a linear combination of the source symbols stored in it (i.e. $\mathcal{L}_{(4,1)}\{\mathcal{X}_2\}$) and transfers it to the newcomer. This is nothing but the additional symbol and the regeneration is complete. One can check that any of the systematic nodes can be regenerated similarly. Now, consider that the last node is failed and the newcomer talks to nodes 1, 2, 3, and 4. Each systematic node i , $1 \leq i \leq k$, gives random linear combination of the source symbols stored in it, i.e., $\mathcal{L}_{(i,6)}\{\mathcal{X}_i\}$. The parity node 4 gives random linear combination of all the symbols stored in it.

Let us now consider the reconstruction. Suppose the DC connects to nodes 1, 4, and 5. Symbol set \mathcal{X}_1 can be directly downloaded from node 1 and it remains to reconstruct sets \mathcal{X}_2 and \mathcal{X}_3 . It can be observed that \mathcal{X}_2 can be decoded from $y_{(4,1)}$, $y_{(2,4)}$, and $y_{(2,5)}$. To decode \mathcal{X}_3 , we have two explicit linear equations in the form of $y_{(3,4)}$ and $y_{(3,5)}$. The third linear combination in \mathcal{X}_3 can be obtained by using one of the two globally combined parity symbols ($y_{(4,4)}$ or $y_{(4,5)}$), and \mathcal{X}_3 can be decoded.³ In similar way, it is possible to reconstruct the entire file by connecting to any k nodes.

Now, let us consider unequal additional symbols case. Let $k = 4$, $n = 6$, and $d = 5$. For $\beta = 1$, this gives $B = 14$ and $\alpha = 5$. Equation (7) below shows the code construction for this case in the matrix form.

In this case, we form the subsets in the similar way as follows: $\mathcal{X}_1 = \{x_1, x_5, x_9, x_{13}\}$, $\mathcal{X}_2 = \{x_2, x_6, x_{10}, x_{14}\}$, $\mathcal{X}_3 = \{x_3, x_7, x_{11}\}$, and $\mathcal{X}_4 = \{x_4, x_8, x_{12}\}$. The additional symbols are linear combinations of the systematic symbols stored in cyclically next (or previous) storage node. The regeneration and reconstruction processes are same as in equal

³It is required that the implicit combination obtained from one of the globally combined parity symbols is linearly independent of the explicit combinations available, for which we rely on high field size.

additional symbols case.

3.2 Generic Construction for MBR point

Let us consider generalized construction for MBR point. Let the parameters k , n , d , and β be selected. Parameters B and α are obtained from (4) and (5), respectively. As stated previously, we assume $\beta = 1$ and the codes for higher β can be obtained by concatenating the ones for $\beta = 1$. Consider the following two cases.

3.2.1 Unequal Additional Symbols

As seen in the example, in this case, half the systematic nodes need to store $\lfloor \frac{(k-1)}{2} \rfloor$ additional symbols whereas remaining half need to store $\lceil \frac{(k-1)}{2} \rceil$ additional symbols. The code construction is as follows.

Divide the set of source symbols \mathcal{X} into k subsets with as follows. First $\frac{k}{2}$ subsets contain $\lceil \frac{B}{k} \rceil$ symbols whereas next $\frac{k}{2}$ subsets contain $\lfloor \frac{B}{k} \rfloor$ symbols.

$$\begin{aligned} \mathcal{X}_j &= \left\{ x_j, x_{j+k}, \dots, x_{j+(\lceil \frac{B}{k} \rceil - 1)k} \right\}, & j = 1, 2, \dots, \frac{k}{2} \\ \mathcal{X}_j &= \left\{ x_j, x_{j+k}, \dots, x_{j+(\lfloor \frac{B}{k} \rfloor - 1)k} \right\}, & j = \frac{k}{2} + 1, \dots, k \end{aligned} \quad (8)$$

- *Systematic Symbols*

$$y_{(i,j)} = x_{(i-1)k+j}, \quad (9)$$

for

$$\begin{aligned} j &= 1, 2, \dots, k, \\ i &= \begin{cases} 1, 2, \dots, \lceil \frac{B}{k} \rceil & \text{if } j \leq \frac{k}{2} \\ 1, 2, \dots, \lfloor \frac{B}{k} \rfloor & \text{if } j > \frac{k}{2}. \end{cases} \end{aligned}$$

- *Additional Symbols*

$$y_{(\lceil \frac{B}{k} \rceil + l, j)} = \mathcal{L}_{(\lceil \frac{B}{k} \rceil + l, j)} \left\{ \mathcal{X}_{((j+l-1) \bmod k) + 1} \right\}, \quad (10)$$

for

$$1 \leq j \leq \frac{k}{2}, 1 \leq l \leq \alpha - \left\lceil \frac{B}{k} \right\rceil.$$

$$y_{(\lfloor \frac{B}{k} \rfloor + l, j)} = \mathcal{L}_{(\lfloor \frac{B}{k} \rfloor + l, j)} \left\{ \mathcal{X}_{((j+l-1) \bmod k) + 1} \right\}, \quad (11)$$

for

$$\frac{k}{2} + 1 \leq j \leq k, 1 \leq l \leq \alpha - \left\lfloor \frac{B}{k} \right\rfloor.$$

- *Parity Symbols*

$$y_{(i,j)} = \mathcal{L}_{(i,j)} \left\{ \mathcal{X}_{((i-1) \bmod k) + 1} \right\}, \quad (12)$$

for

$$1 \leq i \leq k, k+1 \leq j \leq n.$$

- *Global Parity Symbols*

$$y_{(i,j)} = \mathcal{L}_{(i,j)} \{\mathcal{X}\}, \quad (13)$$

for

$$k+1 \leq i \leq \alpha, k+1 \leq j \leq n.$$

Note that in case of $d = k$, global parity symbols are not required.

$$\mathbf{Y}_{(6,3,4)} = \begin{bmatrix} x_1 & x_2 & x_3 & \mathcal{L}_{(1,4)}\{\mathcal{X}_1\} & \mathcal{L}_{(1,5)}\{\mathcal{X}_1\} & \mathcal{L}_{(1,6)}\{\mathcal{X}_1\} \\ x_4 & x_5 & x_6 & \mathcal{L}_{(2,4)}\{\mathcal{X}_2\} & \mathcal{L}_{(2,5)}\{\mathcal{X}_2\} & \mathcal{L}_{(2,6)}\{\mathcal{X}_2\} \\ x_7 & x_8 & x_9 & \mathcal{L}_{(3,4)}\{\mathcal{X}_3\} & \mathcal{L}_{(3,5)}\{\mathcal{X}_3\} & \mathcal{L}_{(3,6)}\{\mathcal{X}_3\} \\ \mathcal{L}_{(4,1)}\{\mathcal{X}_2\} & \mathcal{L}_{(4,2)}\{\mathcal{X}_3\} & \mathcal{L}_{(4,3)}\{\mathcal{X}_1\} & \mathcal{L}_{(4,4)}\{\mathcal{X}\} & \mathcal{L}_{(4,5)}\{\mathcal{X}\} & \mathcal{L}_{(4,6)}\{\mathcal{X}\} \end{bmatrix} \quad (6)$$

$$\mathbf{Y}_{(6,4,5)} = \begin{bmatrix} x_1 & x_2 & x_3 & x_4 & \mathcal{L}_{(1,5)}\{\mathcal{X}_1\} & \mathcal{L}_{(1,6)}\{\mathcal{X}_1\} \\ x_5 & x_6 & x_7 & x_8 & \mathcal{L}_{(2,5)}\{\mathcal{X}_2\} & \mathcal{L}_{(2,6)}\{\mathcal{X}_2\} \\ x_9 & x_{10} & x_{11} & x_{12} & \mathcal{L}_{(3,5)}\{\mathcal{X}_3\} & \mathcal{L}_{(3,6)}\{\mathcal{X}_3\} \\ x_{13} & x_{14} & \mathcal{L}_{(4,3)}\{\mathcal{X}_1\} & \mathcal{L}_{(4,4)}\{\mathcal{X}_2\} & \mathcal{L}_{(4,5)}\{\mathcal{X}_4\} & \mathcal{L}_{(4,6)}\{\mathcal{X}_4\} \\ \mathcal{L}_{(5,1)}\{\mathcal{X}_2\} & \mathcal{L}_{(5,2)}\{\mathcal{X}_3\} & \mathcal{L}_{(5,3)}\{\mathcal{X}_4\} & \mathcal{L}_{(5,4)}\{\mathcal{X}_1\} & \mathcal{L}_{(5,5)}\{\mathcal{X}\} & \mathcal{L}_{(5,6)}\{\mathcal{X}\} \end{bmatrix} \quad (7)$$

3.2.2 Equal Additional Symbols

It can be observed that in this case $\frac{B}{k}$ is an integer and the cardinality of all the subsets is the same. Construction of all the types of symbols remains the same.

In both equal and unequal additional symbols cases, to regenerate a failed systematic node j , the newcomer downloads $\beta = 1$ symbol each from d nodes forming a set \mathcal{D}_j . Any node $l \in \mathcal{D}_j$ passes either a coded symbol $y_{(j,l)}$ or a random linear combination of the symbols stored in it along with the associated encoding vector. The reconstruction can be carried out by connecting to any k nodes.

Now, in the following theorems it is proved that the above construction is optimal for regeneration as well as reconstruction.

THEOREM 1. *For an MBR code constructed as mentioned above, it is possible to regenerate a failed storage node j under hybrid model by downloading β symbols from any set of d nodes \mathcal{D}_j , provided \mathcal{D}_j includes all the surviving systematic nodes.*

PROOF. See Appendix 7. \square

THEOREM 2. *In an MBR code constructed as mentioned above, it is possible to reconstruct the B source symbols by connecting to any k out of n nodes.*

PROOF. See Appendix 8. \square

3.3 Comparison with the Product-Matrix Framework

In this section, we compare the proposed construction with the product-matrix framework presented in [13, 5]. We begin by briefly explaining the product-matrix framework. Under this framework, the code is characterized by an $n \times \alpha$ code matrix $\mathbf{C} = \mathbf{\Psi}\mathbf{M}$. Here, $\mathbf{\Psi}$ is an $n \times d$ encoding matrix, and \mathbf{M} is a $d \times \alpha$ message matrix which contains some permutation of B data symbols with possible repetition of certain symbols (see [5] for details). The j th row of \mathbf{C} contains the symbols to be stored by the j th node. Also, each node j is associated with a distinct encoding vector given by the j th row of $\mathbf{\Psi}$ denoted by Ψ_j .

During the regeneration of j th failed node, the newcomer connects to any d surviving nodes, referred to as helping nodes, and downloads $\beta = 1$ symbol from each one. Each of the d nodes passes the inner product of all the α symbols stored in it with a projection vector μ_j which contains a subset of components of Ψ_j . During the reconstruction, the DC connects to any k nodes and downloads all the symbols stored in them and then decodes the data symbols.

Let us now highlight the differences between product-matrix framework and the proposed framework. First difference is in the number of symbols read by each of the helping

nodes during regeneration. Observe that during regeneration, the product-matrix framework passes on the projection of all the symbols stored in each of the helper nodes. This necessitates reading all the symbols stored in each of the helping nodes. On the contrary, in case of proposed network coding based framework, only $\lceil \frac{k-1}{2} \rceil$ nodes have to read all of their symbols during the regeneration of a systematic node. During the regeneration of a parity node, the k systematic nodes need to read at most $\lceil B/k \rceil$ symbols which is strictly less than α for MBR case.

Secondly, the systematic nodes in case of the product-matrix framework store some permutation of the original data symbols. Therefore, even if the DC connects to systematic nodes, it has to rearrange the downloaded symbols so as to reconstruct the original file. On the contrary, in case of proposed framework, the systematic nodes store the original symbols sequentially. Thus, when the DC connects to all the systematic nodes, reconstructing the file is very simple.

Third difference is in the type of data passed by the newcomer to the helping nodes. In product-matrix framework, each of the helping nodes should know the coding-vector associated with the failed node. Thus, newcomer passes the coding vector of the failed node to each of the helping nodes. On the other hand, the proposed framework requires that a newcomer passes only the index of the failed node, which would decide which symbols to pass.

Lastly, the difference lies in the number and dimension of the encoding vectors stored by each of the storage nodes. In case of the proposed scheme, each node should store α coding vectors each of dimension B . On the contrary, in product-matrix framework, each node stores only one $d \times 1$ encoding vector. In this case, product-matrix constructions are beneficial.

4. ANALYSIS

4.1 Subspace Properties

In this section, we show that the proposed constructions satisfy the necessary subspace properties for a linear exact regenerating code.

A linear exact regenerating code can be characterized using a subspace based approach as stated in [10, 5]. It is worth recollecting from section 2 that each node stores αB encoding coefficients corresponding to α coded symbols stored on it. Let $\mathbf{c}_{(i,j)} = [c_{(i,j)}^1, c_{(i,j)}^2, \dots, c_{(i,j)}^B]$ be a code vector corresponding to symbol $y_{(i,j)}$. Then, the j th node stores α code vectors $\mathbf{c}_{(1,j)}, \mathbf{c}_{(2,j)}, \dots, \mathbf{c}_{(\alpha,j)}$ corresponding to the α symbols stored on it. Note that these code vectors can be used

to define the code, as the linear combinations corresponding to code symbols are specified by these vectors. Also, linear operations performed on the stored symbols are equivalent to the same operations performed on these vectors. Thus, it can be considered that each node stores a subspace of dimension at most α [10]. If \mathbf{S}_j denotes subspace stored on node j , we have

$$\mathbf{S}_j = \langle \mathbf{c}_{(1,j)}, \mathbf{c}_{(2,j)}, \dots, \mathbf{c}_{(\alpha,j)} \rangle, \quad 1 \leq j \leq n, \quad (14)$$

where $\langle \cdot \rangle$ denotes span of vectors.

The subspace based approach can be used to state some necessary conditions that linear exact MBR and MSR codes should satisfy. Subsequent lemmas show that the proposed construction satisfies the necessary conditions for MBR point as stated in [10].

LEMMA 1. *For an MBR code constructed as given in section 3.2, each node stores an α -dimensional subspace, i.e., $\dim \{\mathbf{S}_j\} = \alpha$ for $1 \leq j \leq n$.*

PROOF. See Appendix 9. \square

LEMMA 2. *For an MBR code constructed as given in section 3.2, the intersection of the subspace stored on node j that is to be regenerated with the subspace stored on any other node has dimension β , i.e., $\dim \{\mathbf{S}_j \cap \mathbf{S}_i\} = \beta$, $i \neq j$. Also, the intersection subspaces of node j with the d nodes participating in regeneration are linearly independent, i.e., subspaces $\mathbf{S}_j \cap \mathbf{S}_i$ are linearly independent $\forall i \in \mathcal{D}_j$.*

PROOF. See Appendix 10. \square

4.2 Effect of Field Size

Now, let us consider the effect of field size on the probability of successful regeneration and reconstruction. Our aim in this section is to present analytical expressions which will serve as the guidelines to practically select the appropriate field size. In case of regeneration, we characterize the lower bound on the probability of success. Further, for some combinations of parameters we derive only the approximate results which will satisfy our purpose of providing the guidelines for selecting field size.

THEOREM 3. *For an MBR code constructed as given in section 3.2, the probability of successful regeneration of a systematic node is given as*

$$p_{reg}^{MBR} = \prod_{i=0}^{\delta-1} \left(1 - \frac{1}{q^{\delta-i}}\right) \left[\left(1 - \frac{1}{q}\right)\right]^{\alpha-\delta} \quad (15)$$

where $\delta = |\mathcal{X}_j| = \binom{B}{k}$ when k is odd, while $\delta = \lceil \frac{B}{k} \rceil$ for first $\frac{k}{2}$ systematic nodes and $\delta = \lfloor \frac{B}{k} \rfloor$ for last $\frac{k}{2}$ systematic nodes when k is even.

The probability of successful reconstruction can be lower bounded as

$$p_{rec}^{MBR} \geq p_1 p_2, \quad \text{if } d > \left\lceil \frac{3k-1}{2} \right\rceil \text{ and } k \text{ is odd} \quad (16)$$

$$\gtrsim p_1 p_2, \quad \text{if } d > \left\lceil \frac{3k-1}{2} \right\rceil \text{ and } k \text{ is even} \quad (17)$$

$$\geq p_3, \quad \text{if } d \leq \left\lceil \frac{3k-1}{2} \right\rceil \text{ and } k \text{ is odd} \quad (18)$$

$$\geq p_3 p_4, \quad \text{if } d \leq \left\lceil \frac{3k-1}{2} \right\rceil \text{ and } k \text{ is even}, \quad (19)$$

where

$$p_1 = \left[\prod_{i=0}^{k-1} \left(1 - \frac{1}{q^{\delta-i}}\right) \right]^k, \quad (20)$$

$$p_2 = \prod_{i=0}^{B-k^2-1} \left(1 - \frac{1}{q^{B-k^2-i}}\right), \quad (21)$$

$$p_3 = \left[\prod_{i=0}^{\delta-1} \left(1 - \frac{1}{q^{\delta-i}}\right) \right]^k, \quad (22)$$

$$p_4 = \prod_{i=0}^{\frac{k}{2}-1} \left(1 - \frac{1}{q^{\frac{k}{2}-i}}\right). \quad (23)$$

PROOF. See Appendix 11. \square

4.3 Numerical Results on the Effect of the Field Size

In this section we study the effect of the Galois field size by carrying out the numerical simulations. Since the code design is independent of the total number of storage nodes n , the parameters under consideration are k and d . For a fixed k , we first choose d close to k and then much greater than k .

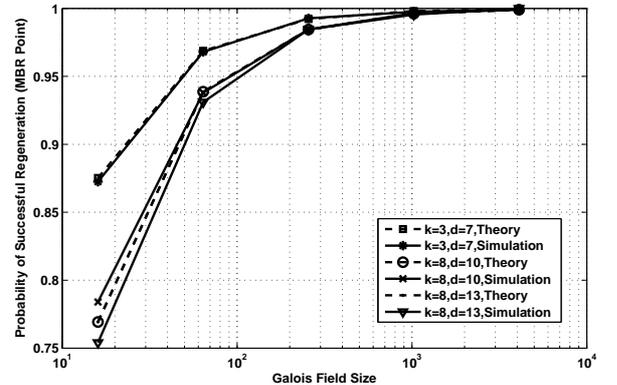


Figure 3: Probability of successful regeneration vs Galois field size at MBR point

For MBR point, Fig. 3 shows the probability of successful regeneration p_{reg}^{MBR} versus field size q for various values of k and d . Note that the analytical curves for $k = 8$, $d = 10$ and $k = 8$, $d = 13$ are overlapping. The increase in the field size enhances the probability that the random linear combinations are linearly independent and thus p_{reg}^{MBR} also increases. It can be seen that the analytical and simulation results are in close agreement for all combinations of k and d . Further, one can observe that p_{reg}^{MBR} is higher for smaller value of k . This is because the number of additional symbols increases with k , and these symbols need to be linearly independent of the additional symbols that are already present in the system. Also, increase in d increases $\delta = |\mathcal{X}_j|$ and reduces p_{reg}^{MBR} (see (15)).

For MBR point, the probability of successful reconstruction p_{rec}^{MBR} versus the field size for various values of k and d is shown in Fig. 4. Note that the curves for $k = 7$, $d = 14$ and $k = 8$, $d = 16$ are almost overlapping. It is apparent that the analytical results closely match the simulations. The dip

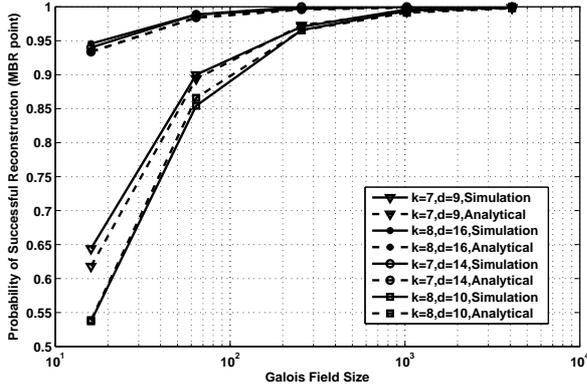


Figure 4: Probability of successful reconstruction vs Galois field size at MBR point

in the p_{rec}^{MBR} for $k = 7$ and $d = 9$ is due to the fact that, in this case, submatrices of smaller size ($\delta \times \delta = 6 \times 6$) need to be invertible and the probability of this event is less at the smaller field size. From simulation results at various k and d , it can be considered that, unless δ is very small, p_{rec}^{MBR} is almost independent of the parameters k and d for given q .

5. FEW REMARKS ON USING THE PROPOSED FRAMEWORK FOR THE MSR CASE

It is possible to extend the proposed framework for MSR point as well. However, we note that for MSR point, the structure of the proposed code is very similar to that given in [14, 6]. In this section, we describe how the proposed construction can be extended for MSR point.

Now, for MSR point, optimal α and β can be shown as [9]:

$$(\alpha_{MSR}, \beta_{MSR}) = \left(\frac{B}{k}, \frac{B}{dk - k^2 + k} \right) \quad (24)$$

Therefore,

$$B = \beta k(d - k + 1) \quad (25)$$

and

$$\alpha = \beta(d - k + 1). \quad (26)$$

5.1 Example for MSR code

Let $k = 3$, $n = 7$, $d = 6$, and $\beta = 1$. This gives $B = 12$ and $\alpha = 4$. The code construction is shown in (27) (depicted on the next page) in the form of a matrix $\mathbf{Y}_{(n,k,d)}$, where j th column of the matrix represents the elements to be stored on node j .

First, the set \mathcal{X} containing all source symbols is divided into $k = 3$ subsets as follows: $\mathcal{X}_1 = \{x_1, x_4, x_7, x_{10}\}$, $\mathcal{X}_2 = \{x_2, x_5, x_8, x_{11}\}$, and $\mathcal{X}_3 = \{x_3, x_6, x_9, x_{12}\}$. The symbols in set \mathcal{X}_j are stored in node j . Further, the subsets \mathcal{X}_{j+k} ($1 \leq j \leq k$) are formed by combining the j th symbol in all the nodes except the j th one, i.e., $\mathcal{X}_4 = \{x_2, x_3\}$, $\mathcal{X}_5 = \{x_4, x_6\}$, and $\mathcal{X}_6 = \{x_7, x_8\}$. Then, the sets $\mathcal{X}_{j+2k} = \mathcal{X}_j \cup \mathcal{X}_{j+k}$, $1 \leq j \leq k$, are constructed, giving \mathcal{X}_7 to \mathcal{X}_9 .

For the non-systematic node j , the i th parity symbol ($1 \leq i \leq k$) is obtained by randomly combining the source

symbols in set \mathcal{X}_{i+2k} , i.e., $y_{(i,j)} = \mathcal{L}_{(i,j)} \{\mathcal{X}_{i+2k}\}$. The remaining parity symbols, i.e. globally combined parity symbols, are random combinations of all the source symbols, giving $y_{(i,j)} = \mathcal{L}_{(i,j)} \{\mathcal{X}\}$. Similar to MBR case, it is assumed that all the coefficient vectors corresponding to linear combinations are linearly independent of each other, which can be achieved by selecting the arbitrarily high field size [12].

Now, let us consider the regeneration of a failed systematic node, assuming that the second node is failed. Systematic nodes 1 and 3, give the symbols of set \mathcal{X}_5 and parity nodes 4 to 7 give linear combinations in \mathcal{X}_8 . With $\mathcal{X}_8 = \mathcal{X}_2 \cup \mathcal{X}_5$ and \mathcal{X}_5 known, it is equivalent to getting four combinations in \mathcal{X}_2 , from which elements of \mathcal{X}_2 can be decoded. Any of the systematic nodes can be regenerated similarly.

Let us consider the reconstruction. Suppose the DC connects to nodes 1, 5, and 7. Sets \mathcal{X}_2 and \mathcal{X}_3 are to be decoded, as set \mathcal{X}_1 can be directly downloaded from node 1. Since parity symbols $y_{(1,5)}$ and $y_{(1,7)}$ are $\mathcal{L}\{\mathcal{X}_7\} = \mathcal{L}\{\mathcal{X}_1 \cup \mathcal{X}_5\}$, and \mathcal{X}_1 is known, it is possible to decode \mathcal{X}_5 . After substituting \mathcal{X}_1 and \mathcal{X}_5 in globally combined parity symbols, we have six equations in six unknowns which can be easily solved, under the assumption of high field size producing linearly independent equations. It is possible to reconstruct the entire file in similar way by connecting to any other k nodes.

5.2 Generic Code Construction for MSR Point

Let k , n , d , and β be fixed. Then, B and α can be given as (25) and (26), respectively. We present the construction for $\beta = 1$, as the code for higher β can be obtained by concatenation, similar to MBR case. The construction is given as follows.

Divide the set of source symbols \mathcal{X} into following subsets:

$$\begin{aligned} \mathcal{X}_j &= \left\{ x_j, x_{j+k}, \dots, x_{j+\left(\frac{B}{k}-1\right)k} \right\}, \\ \mathcal{X}_{j+k} &= \left\{ x_{(j-1)k+1}, x_{(j-1)k+2}, \dots, x_{jk} \right\} \setminus x_{(j-1)k+j} \quad (28) \\ \mathcal{X}_{j+2k} &= \mathcal{X}_j \cup \mathcal{X}_{j+k}, \end{aligned}$$

where $j = 1, 2, \dots, k$ and '\setminus' denotes subtraction of sets.

- *Systematic Symbols*

$$y_{(i,j)} = x_{(i-1)k+j}, \quad (29)$$

for

$$\begin{aligned} j &= 1, 2, \dots, k, \\ i &= 1, 2, \dots, \frac{B}{k}. \end{aligned}$$

- *Parity Symbols*

$$y_{(i,j)} = \mathcal{L}_{(i,j)} \left\{ \mathcal{X}_{((i-1) \bmod k)+1+2k} \right\}, \quad (30)$$

for

$$1 \leq i \leq k, k+1 \leq j \leq n.$$

- *Global Parity Symbols*

$$y_{(i,j)} = \mathcal{L}_{(i,j)} \{\mathcal{X}\}, \quad (31)$$

for

$$k+1 \leq i \leq \alpha, k+1 \leq j \leq n.$$

In case of $d = 2k - 1$, global parity symbols are not required.

To regenerate a failed systematic node j , a node $l \in \mathcal{D}_j$ passes symbol $y_{(j,l)}$ and associated code vector. The reconstruction is possible by connecting to any k nodes.

$$\mathbf{Y}_{(7,3,6)} = \begin{bmatrix} x_1 & x_2 & x_3 & \mathcal{L}_{(1,4)} \{\mathcal{X}_7\} & \mathcal{L}_{(1,5)} \{\mathcal{X}_7\} & \mathcal{L}_{(1,6)} \{\mathcal{X}_7\} & \mathcal{L}_{(1,7)} \{\mathcal{X}_7\} \\ x_4 & x_5 & x_6 & \mathcal{L}_{(2,4)} \{\mathcal{X}_8\} & \mathcal{L}_{(2,5)} \{\mathcal{X}_8\} & \mathcal{L}_{(2,6)} \{\mathcal{X}_8\} & \mathcal{L}_{(2,7)} \{\mathcal{X}_8\} \\ x_7 & x_8 & x_9 & \mathcal{L}_{(3,4)} \{\mathcal{X}_9\} & \mathcal{L}_{(3,5)} \{\mathcal{X}_9\} & \mathcal{L}_{(3,6)} \{\mathcal{X}_9\} & \mathcal{L}_{(3,7)} \{\mathcal{X}_9\} \\ x_{10} & x_{11} & x_{12} & \mathcal{L}_{(4,4)} \{\mathcal{X}\} & \mathcal{L}_{(4,5)} \{\mathcal{X}\} & \mathcal{L}_{(4,6)} \{\mathcal{X}\} & \mathcal{L}_{(4,7)} \{\mathcal{X}\} \end{bmatrix} \quad (27)$$

6. CONCLUSION AND FUTURE WORK

In this article, we presented network coding based constructions for the systematic regenerating codes at Minimum Bandwidth Regenerating (MBR) point. The codes have several significant features: (a) the codes are systematic, i.e., the data symbols are always present in the uncoded form; (b) there is flexibility in choosing the number of nodes that would participate in the regeneration of a failed node; (c) the constructions are independent of the total number of nodes, making it possible to add or remove any number of parity node. These features make the codes reconfigurable and amenable for practical usage. The article also presented rigorous analysis of the effect of Galois field size on the success of regeneration and reconstruction for the proposed codes. This study provides some useful guidelines for carefully selecting the field size for the practical implementations.

Furthermore, we observed that the proposed framework can be extended to Minimum Storage Regenerating (MSR) point, which results in MSR codes having similar structure to the well known codes [14, 6]. Indeed, this similarity depicts the versatility of the proposed network coding based framework and opens up an interesting future direction to bring out the analytical similarities between our framework and that of [14, 6]. It would also be interesting to rigorously prove the regeneration and reconstruction properties of the proposed framework at MSR point.

Other promising future directions would be to (a) examine whether deterministic code constructions can be designed under this framework; (b) practically implement the proposed codes in a distributed storage system and compare its performance with the existing regenerating codes.

APPENDIX

In this appendix, we provide the proofs for the lemmas and the theorems stated in the article.

7. PROOF OF THEOREM 1

We prove the theorem for $\beta = 1$, as for $\beta > 1$, the construction can be considered as concatenation of codes with $\beta = 1$. Now, when a systematic (or parity) node fails, there are α unknowns to be determined by downloading one symbol each from $d = \alpha$ nodes out of which $k - 1$ (or k) nodes are systematic.

Consider the failure of a systematic node. In equal additional symbols case, out of α unknowns, $\alpha - \frac{k-1}{2}$ symbols are systematic symbols and remaining $\frac{k-1}{2}$ are additional symbols. The additional symbols can be replaced by their functional equivalents obtained by randomly combining source symbols from cyclically next (or previous) $\frac{k-1}{2}$ systematic nodes. The coded symbols along with the corresponding code vectors downloaded from the remaining systematic and parity nodes form $\alpha - \frac{k-1}{2}$ linear equations, which can be solved under high field size assumption, to obtain $\alpha - \frac{k-1}{2}$ systematic symbols. Exactly similar arguments hold for re-

pair of systematic node in case of unequal additional symbols, with only difference in number of additional symbols.

When a parity node is repaired, the each of the k systematic nodes functionally generates a parity symbols by randomly combining the data symbols stored on it. Each of the $d - k$ parity nodes participating in the repair pass random linear combination of all the symbols stored in it. This functionally repairs the $d - k$ globally combined parity symbols.

Note that one can question whether these global parity symbols create some linear dependencies across parity nodes which can potentially harm the reconstruction process. We prove in theorem 2 that it does not.

8. PROOF OF THEOREM 2

It can be observed that reconstructing the file is equivalent to solving for B unknowns using $k\alpha = kd$ equations. Here, we can consider systematic symbols as trivial equations. Now, we show that the number of parity symbols (which essentially is the number of equations) is equal to the number of unknowns. Assume that the DC connects to s systematic nodes and p parity nodes (note, $s + p = k$).

Let us first consider the case of equal additional symbols ($k : \text{odd}$). In this case, each systematic node has $\frac{(k-1)}{2}$ additional symbols and $\frac{(2d-k+1)}{2}$ data symbols. When DC visits s systematic nodes, it directly downloads $s \frac{(2d-k+1)}{2}$ data symbols (trivial equations). Therefore, number of remaining unknowns is $B - s \frac{(2d-k+1)}{2} = p \frac{(2d-k+1)}{2}$, since $B = k \frac{(2d-k+1)}{2}$. Now, the total number of additional symbols obtained from s nodes is $s \frac{(k-1)}{2}$.

First, we compute the number of *useful* additional symbols, i.e., the additional symbols which does not correspond to the data symbols present in the s systematic nodes visited. (Note that if data symbols which are combined to generate an additional symbol have already been downloaded, then that additional symbol turns out to be *useless*.)

CLAIM 1. *For equal additional symbols case, if a DC connects to s systematic nodes, then the number of "useful" additional symbols is at least $\frac{s}{2}(k - s)$.*

PROOF. Note that by construction, each additional symbol is a random linear combination of data symbols stored in cyclically next (or previous) systematic node. Therefore, if DC connects to cyclically sequential systematic nodes, least number of additional symbols will be useful. Thus, we consider this case.

Now, if we consider the set of first additional symbol in each of the s systematic nodes, $(s - 1)$ additional symbols in this set are useless as they correspond to data symbols stored on cyclically next $(s - 1)$ nodes. From the set of second additional symbols in each of the s systematic nodes, $(s - 2)$ symbols are useless, and so on. Also, note that the the total number of additional symbols obtained from s systematic nodes is $s \frac{(k-1)}{2}$. Therefore, number of useful

additional symbols is at least $s \frac{(k-1)}{2} - [(s-1) + (s-2) + \dots + 1] = \frac{s}{2}(k-s)$. \square

Now, let us focus on the parity nodes. There are total $p\alpha = pd$ parity symbols, among which the pk parity symbols correspond to data symbols stored in each of the k systematic nodes, while rest $p(d-k)$ are globally combined parity symbols. Out of pk parity symbols, ps symbols correspond to s systematic nodes already visited by DC and thus are useless. Hence, number of useful (local) parity symbols are $(pk - ps)$.

Last part is to count the number of useful global parity symbols. Note that after the regeneration of a failed parity node, any repaired global parity symbol is a random linear combination of all the symbols in some other parity node that participated in the regeneration. If that particular parity node is in the set of p parity nodes visited by DC, this global parity symbol will be useless. Now, assume the worst case that DC has visited p parity nodes such that each one participated in the regeneration of the other. Without loss of generality, assume that these p nodes are cyclically sequential. Then, following on the similar lines as with systematic nodes, we can see that the number of useful global parity symbols is at least $p(d-k) - [(p-1) + (p-2) + \dots + 1] = p(d-k) - \frac{p(p-1)}{2}$.

Now, total number of *useful* equations is at least $\frac{s}{2}(k-s)$ additional symbols + $(pk - ps)$ parity symbols + $p(d-k) - \frac{p(p-1)}{2}$ global parity symbols. It is straightforward to show that this equals $\frac{p(2d-k+1)}{2}$, which are the number of unknown data symbols.

For unequal additional symbols case (even k), we follow the exact similar lines to show that the number of unknowns equals the number of useful equations. The details are omitted due to their similarity with the presented case.

Thus, the construction always guarantees sufficient number of equations (parity symbols), and under the assumption of linear independence through high field size, the file can be reconstructed.

9. PROOF OF LEMMA 1

From (10), it is easy to observe that for any systematic node, the additional symbols are linearly independent of the source symbols stored in that node. The α linearly independent symbols form an α dimensional subspace.

In case of parity nodes, the underlying assumption in the construction is that the linear combinations given by (12) and (13) are linearly independent which can be achieved by choosing sufficiently high field size.

10. PROOF OF LEMMA 2

The source symbols stored on the j th systematic node (excluding the symbols that are replicated) form the subset \mathcal{X}_j . If we consider any parity node i , $k+1 \leq i \leq n$, it contains β linear combinations in \mathcal{X}_j . The other parity symbols are either formed using sets \mathcal{X}_l , $l \neq j$, or are globally combined parity symbols in \mathcal{X} . Thus, the dimension of the intersection of subspaces is β .

If we consider a systematic node i , $i \neq j$, it can be observed from the construction that either node i contains $\mathcal{L}\{\mathcal{X}_j\}$ or node j contains $\mathcal{L}\{\mathcal{X}_j\}$ for $\beta = 1$. Since the case of higher β can be obtained by concatenation of $\beta = 1$ codes, we can see that dimension of intersection of subspaces

of systematic node j with any other systematic node i , $i \neq j$, is β .

The linear independence of the intersection subspaces is implicit in the construction through the assumption of arbitrarily high field size.

11. PROOF OF THEOREM 3

We divide the proof into two parts. **1. Proof for regeneration:**

Throughout the proof, we use the following expression for the probability that a $r \times c$ matrix ($r \leq c$) with each element chosen uniformly and independently over $GF(q)$ has a rank ρ ($\rho \leq r$) [15]:

$$p = \frac{\prod_{i=0}^{\rho-1} (q^r - q^i) \prod_{i=0}^{\rho-1} (q^c - q^i)}{q^{rc} \prod_{i=0}^{\rho-1} (q^\rho - q^i)}. \quad (32)$$

Now, to regenerate a failed node j , we need to find out α unknowns, out of which $\delta = |\mathcal{X}_j|$ are source symbols and remaining are additional symbols. The probability of successful regeneration is (i) the probability that we obtain δ elements in \mathcal{X}_j and (ii) the additional symbol supplied by each l of the $\alpha - \delta$ systematic nodes is linearly independent of any $\delta - 1$ symbols in set \mathcal{X}_l already present in the nodes. From (32), the probability of (i) can be given as

$$p = \prod_{i=0}^{\delta-1} \left(1 - \frac{1}{q^{\delta-i}}\right). \quad (33)$$

The probability that a uniform random vector of dimension δ corresponding to an additional symbol is linearly independent of $\delta - 1$ vectors that are linearly independent is $(1 - 1/q)$. Thus, probability of part (ii) is $\left[\left(1 - \frac{1}{q}\right)\right]^{\alpha-\delta}$. From these, equation (15) follows.

2. Proof for reconstruction:

To characterize the lower bound on reconstruction probability, we consider the case where maximum number of equations are required to be solved. This will be the case when $n \geq 2k$ and DC connects to k parity nodes. Now, in this scenario, we have k equations each in elements of sets \mathcal{X}_j for $1 \leq j \leq k$ and remaining equations in elements of \mathcal{X} .

For each of the four cases, we consider the structure of the coefficient matrix and find the probability that the matrix is invertible. Note that, to obtain the structure, the symbols can be labeled in such a way that first \mathcal{X}_1 symbols form set \mathcal{X}_1 , next \mathcal{X}_1 symbols form set \mathcal{X}_2 and so on. This facilitates to obtain organized structures for coefficient matrices. Also, to simplify the notation, we assume that the parity nodes are numbered from 1 to k .

Case 1: $d > \lceil \frac{3k-1}{2} \rceil$ and equal parity symbols.

In this case, $\delta > k$ and $B = k\delta$. The structure of the $B \times B$ coefficient matrix can be given as

$$\mathbf{C}_{(n,k,d)}^1 = \begin{bmatrix} \tilde{\Lambda} \\ \Lambda \end{bmatrix}, \quad (34)$$

where,

$$\tilde{\Lambda} = \begin{bmatrix} \Lambda_1 & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \Lambda_2 & \dots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \dots & \Lambda_k \end{bmatrix}, \quad (35)$$

with

$$\Lambda_1 = \begin{bmatrix} c_{(1,1)}^1 & c_{(1,1)}^2 & \cdots & c_{(1,1)}^\delta \\ c_{(1,2)}^1 & c_{(1,2)}^2 & \cdots & c_{(1,2)}^\delta \\ \vdots & \vdots & \ddots & \vdots \\ c_{(1,k)}^1 & c_{(1,k)}^2 & \cdots & c_{(1,k)}^\delta \end{bmatrix} \quad (36)$$

$$\Lambda_2 = \begin{bmatrix} c_{(2,1)}^{\delta+1} & c_{(2,1)}^{\delta+2} & \cdots & c_{(2,1)}^{2\delta} \\ c_{(2,2)}^{\delta+1} & c_{(2,2)}^{\delta+2} & \cdots & c_{(2,2)}^{2\delta} \\ \vdots & \vdots & \ddots & \vdots \\ c_{(2,k)}^{\delta+1} & c_{(2,k)}^{\delta+2} & \cdots & c_{(2,k)}^{2\delta} \end{bmatrix} \quad (37)$$

$$\Lambda_k = \begin{bmatrix} c_{(k,1)}^{(k-1)\delta+1} & c_{(k,1)}^{(k-1)\delta+2} & \cdots & c_{(k,1)}^{k\delta} \\ c_{(k,2)}^{(k-1)\delta+1} & c_{(k,2)}^{(k-1)\delta+2} & \cdots & c_{(k,2)}^{k\delta} \\ \vdots & \vdots & \ddots & \vdots \\ c_{(k,k)}^{(k-1)\delta+1} & c_{(k,k)}^{(k-1)\delta+2} & \cdots & c_{(k,k)}^{k\delta} \end{bmatrix} \quad (38)$$

and

$$\Lambda = \begin{bmatrix} c_{(k+1,1)}^1 & c_{(k+1,1)}^2 & \cdots & c_{(k+1,1)}^{k\delta} \\ \vdots & \vdots & \ddots & \vdots \\ c_{(k+1,k)}^1 & c_{(k+1,k)}^2 & \cdots & c_{(k+1,k)}^{k\delta} \\ c_{(k+2,1)}^1 & c_{(k+2,1)}^2 & \cdots & c_{(k+2,1)}^{k\delta} \\ \vdots & \vdots & \ddots & \vdots \\ c_{(k+2,k)}^1 & c_{(k+2,k)}^2 & \cdots & c_{(k+2,k)}^{k\delta} \\ c_{(\delta,1)}^1 & c_{(\delta,1)}^2 & \cdots & c_{(\delta,1)}^{k\delta} \\ \vdots & \vdots & \ddots & \vdots \\ c_{(\delta,k)}^1 & c_{(\delta,k)}^2 & \cdots & c_{(\delta,k)}^{k\delta} \end{bmatrix} \quad (39)$$

It can be observed that, for the matrix $\mathbf{C}_{(n,k,d)}$ to be invertible, each $k \times \delta$ submatrix Λ_i , $1 \leq i \leq k$, should have rank k and each of the code vector corresponding to the globally combined parity symbols should be linearly independent of the space spanned by the earlier vectors. Using (32), the probability of the first part is given as

$$p_1 = \prod_{i=0}^{k-1} \left(1 - \frac{1}{q^{\delta-i}}\right). \quad (40)$$

If first k^2 rows of the matrix $\mathbf{C}_{(n,k,d)}^1$ are linearly independent (i.e. each of $\Lambda_1, \dots, \Lambda_k$ is invertible), they span a vector space of dimension k^2 and size q^{k^2} . The probability that the next row vector corresponding to globally combined parity (first row of matrix Λ) avoids this space, assuming the uniform distribution, is given as $(1 - q^{k^2}/q^B)$. Thus, probability that each of the $B - k^2$ code vector corresponding to globally combined parity symbols (each row of Λ) is linearly independent of the space spanned by earlier vectors is

$$p_2 = \prod_{i=0}^{B-k^2-1} \left(1 - \frac{1}{q^{B-k^2-i}}\right). \quad (41)$$

Case 2: $d > \lceil \frac{3k-1}{2} \rceil$ and unequal parity symbols.

In this case, $\delta > k$ and $B = k\delta + k/2$. The structure of the coefficient matrix is given as

$$\mathbf{C}_{(n,k,d)}^2 = \begin{bmatrix} \tilde{\Lambda} \\ \Lambda \end{bmatrix} \quad (42)$$

where $\tilde{\Lambda}$ is given as

$$\tilde{\Lambda} = \begin{bmatrix} \Lambda_1 & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \Lambda_2 & \cdots & \mathbf{0} & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \Lambda_k & \mathbf{0} \end{bmatrix} \quad (43)$$

with $\Lambda_1, \Lambda_2, \dots, \Lambda_k$ same as given in Case 1 and Λ is given by (44).

Now, the probability that each submatrix has rank k is same as (40). To consider linear independence of globally combined code vectors, note that if any of the last $\frac{k}{2}$ entries in a globally combined vector, it can not be constructed by linearly combining the first k^2 vectors. Using such arguments, it may be possible to consider various combinatorial possibilities of globally combined code vectors not being falling in the space spanned by k^2 vectors. However, since we are interested in establishing the guidelines for choosing the field size, we assume k to be small and can build similar arguments as in case 1. Thus, the probability of successful reconstruction can be approximated as in case 1.

Case 3: $d \leq \lceil \frac{3k-1}{2} \rceil$ and equal parity symbols.

In this case, $\delta \leq k$ and $B = k\delta$. The structure of the coefficient matrix is given as

$$\mathbf{C}_{(n,k,d)}^3 = \begin{bmatrix} \Lambda_1 & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \Lambda_2 & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \Lambda_k \end{bmatrix} \quad (45)$$

where $\Lambda_1, \Lambda_2, \dots, \Lambda_k$ have similar structure as in Case 1 except that there are δ rows. For the matrix $\mathbf{C}_{(n,k,d)}$ to be invertible, it is required that each $\delta \times \delta$ submatrix Λ_i , $1 \leq i \leq k$, is invertible. The probability of this event p_4 given by (23) follows directly from (32).

Case 4: $d \leq \lceil \frac{3k-1}{2} \rceil$ and unequal parity symbols.

In this case, $\delta \leq k$ and $B = k\delta + k/2$. The structure of the coefficient matrix is given as

$$\mathbf{C}_{(n,k,d)}^4 = \begin{bmatrix} \tilde{\Lambda} \\ \Lambda \end{bmatrix}, \quad (46)$$

where,

$$\tilde{\Lambda} = \begin{bmatrix} \Lambda_1 & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \Lambda_2 & \cdots & \mathbf{0} & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \Lambda_k & \mathbf{0} \end{bmatrix} \quad (47)$$

in which $\Lambda_1, \Lambda_2, \dots, \Lambda_k$ have similar structure as in Case 1 except that there are δ rows and

$$\Lambda = \begin{bmatrix} c_{(k+1,1)}^1 & c_{(k+1,1)}^2 & \cdots & c_{(k+1,1)}^B \\ c_{(k+1,2)}^1 & c_{(k+1,2)}^2 & \cdots & c_{(k+1,2)}^B \\ \vdots & \vdots & \ddots & \vdots \\ c_{(k+1,k/2)}^1 & c_{(k+1,k/2)}^2 & \cdots & c_{(k+1,k/2)}^B \end{bmatrix} \quad (48)$$

The invertibility of matrix $\mathbf{C}_{(n,k,d)}$ requires that each $\delta \times \delta$ matrix and the last $\frac{k}{2} \times \frac{k}{2}$ matrix is invertible. These probabilities (p_3 and p_4) can be easily found out using (32).

Acknowledgment

The authors would like to thank Prof. P. Vijay Kumar of Indian Institute of Science and Rashmi K. V. and Nihar

$$\Lambda = \begin{bmatrix} c_{(k+1,1)}^1 & c_{(k+1,1)}^2 & \cdots & c_{(k+1,1)}^{k\delta} & c_{(k+1,1)}^{k\delta+1} & \cdots & c_{(k+1,1)}^B \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ c_{(k+1,k)}^1 & c_{(k+1,k)}^2 & \cdots & c_{(k+1,k)}^{k\delta} & c_{(k+1,k)}^{k\delta+1} & \cdots & c_{(k+1,k)}^B \\ c_{(k+2,1)}^1 & c_{(k+2,1)}^2 & \cdots & c_{(k+2,1)}^{k\delta} & c_{(k+2,1)}^{k\delta+1} & \cdots & c_{(k+2,1)}^B \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ c_{(k+2,k)}^1 & c_{(k+2,k)}^2 & \cdots & c_{(k+2,k)}^{k\delta} & c_{(k+2,k)}^{k\delta+1} & \cdots & c_{(k+2,k)}^B \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ c_{(\delta,1)}^1 & c_{(\delta,1)}^2 & \cdots & c_{(\delta,1)}^{k\delta} & c_{(\delta,1)}^{k\delta+1} & \cdots & c_{(\delta,1)}^B \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ c_{(\delta,k)}^1 & c_{(\delta,k)}^2 & \cdots & c_{(\delta,k)}^{k\delta} & c_{(\delta,k)}^{k\delta+1} & \cdots & c_{(\delta,k)}^B \\ c_{(\delta+1,1)}^1 & c_{(\delta+1,1)}^2 & \cdots & c_{(\delta+1,1)}^{k\delta} & c_{(\delta+1,1)}^{k\delta+1} & \cdots & c_{(\delta+1,1)}^B \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ c_{(\delta+1,k/2)}^1 & c_{(\delta+1,k/2)}^2 & \cdots & c_{(\delta+1,k/2)}^{k\delta} & c_{(\delta+1,k/2)}^{k\delta+1} & \cdots & c_{(\delta+1,k/2)}^B \end{bmatrix} \quad (44)$$

Shah of University of California, Berkeley for providing the introduction of the concepts of regenerating codes. Also, the authors would like to thank Dr. B. S. Adiga and Dr. P. Balamuralidhar of TATA Consultancy Services for their useful suggestions and support.

12. REFERENCES

- [1] H. Weatherspoon and J. Kubiatowicz, "Erasure Coding Vs. Replication: A Quantitative Comparison," in *Revised Papers from the First International Workshop on Peer-to-Peer Systems*, ser. IPTPS '01. London, UK: Springer-Verlag, 2002, pp. 328–338.
- [2] A. G. Dimakis, P. B. Godfrey, M. Wainwright, and K. Ramchandran, "Network Coding for Distributed Storage Systems," in *Proc. of IEEE INFOCOM*, Urbana-Champaign, May 2007.
- [3] —, "Network Coding for Distributed Storage Systems," *IEEE Trans. Inf. Theory*, vol. 56, no. 9, pp. 4539–4551, Sep. 2010.
- [4] A. Dimakis, K. Ramchandran, Y. Wu, and C. Suh, "A survey on network codes for distributed storage," *Proc. IEEE*, Mar. 2011.
- [5] K. V. Rashmi, N. B. Shah, P. V. Kumar, and K. Ramchandran, "Optimal Exact-Regenerating Codes for Distributed Storage at the MSR and MBR Points via a Product-Matrix Construction," *IEEE Trans. Inf. Theory*, vol. 57, no. 8, pp. 5227–5239, Aug. 2011.
- [6] C. Suh, , and K. Ramchandran, "Exact regeneration codes for distributed storage repair using interference alignment," *IEEE Trans. Inf. Theory*, vol. 57, no. 3, pp. 1425–1442, Mar. 2011.
- [7] N. B. Shah, K. V. Rashmi, P. V. Kumar, and K. Ramchandran, "Explicit minimizing repair bandwidth for distributed storage," in *Proc. IEEE ITW*, Cairo, Sep. 2009.
- [8] Y. Wu, "A construction of systematic MDS codes with minimum repair bandwidth," *IEEE Trans. Inf. Theory*, vol. 57, no. 6, pp. 3738–3741, Jun. 2011.
- [9] Y. Wu, A. Dimakis, and K. Ramchandran, "Deterministic Regenerating Codes for Distributed Storage," in *Proc. Allerton Conf., Urbana-Champaign*, Sep. 2007.
- [10] K. V. Rashmi, N. B. Shah, P. V. Kumar, and K. Ramchandran, "Explicit construction of optimal exact regenerating codes for distributed storage," in *Proc. Allerton Conf., Urbana-Champaign*, Sep. 2009.
- [11] B. Janakiram, S. Kadhe, and M. G. Chandra, "ExR: A scheme for exact regeneration of a failed node in a distributed storage system," in *Proc. of Int. Conf. on Advances in Distributed and Parallel Computing*, Singapore, Nov. 2010.
- [12] T. Ho, M. Medard, M. Effros, and D. Karger, "On Randomized Network Coding," in *Proc. 41st Allerton Annual Conference on Communication, Control and Computing*, Urbana-Champaign, Oct. 2003.
- [13] K. V. Rashmi, N. B. Shah, P. V. Kumar, and K. Ramchandran, "Explicit and optimal exact-regenerating codes for the minimum-bandwidth point in distributed storage," in *Proc. ISIT*, Austin, Jun. 2010, pp. 1938–1942.
- [14] N. B. Shah, K. V. Rashmi, P. V. Kumar, and K. Ramchandran, "Interference alignment in regenerating codes for distributed storage: Necessity and code constructions," *IEEE Trans. Inf. Theory*, vol. 58, no. 4, pp. 2134–2158, Apr. 2012.
- [15] S. Acedanski, S. Deb, M. Medard, and R. Koetter, "How Good is Random Linear Coding Based Distributed Networked Storage?" in *In NetCod*, 2005.