Repairing Multiple Failures in the Suh-Ramchandran Regenerating Codes

Junyu Chen
Department of Information Engineering
the Chinese University of Hong Kong
Email: cj012@ie.cuhk.edu.hk

Kenneth W. Shum
Institute of Network Coding
the Chinese University of Hong Kong
Email: wkshum@inc.cuhk.edu.hk

***Abstract*— Using the idea of interference alignment, Suh and Ramchandran constructed a class of minimum-storage regenerating codes which can repair one systematic or one parity-check node with optimal repair bandwidth. With the same code structure, we show that in addition to single node failure, double node failures can be repaired collaboratively with optimal repair bandwidth as well. We give an example of how to repair double failures in the Suh-Ramchandran regenerating code with six nodes, and give the proof for the general case.**

***Index Terms*—Distributed storage systems, regenerating codes, interference alignment, super-regular matrix.**

I. Introduction

In a distributed storage system, we encode and distribute a data file of size B to n storage nodes, with two properties that (i) any k nodes are sufficient in rebuilding the original file, and (ii) upon the failure of one or more storage nodes, we can recover the lost information efficiently. Property (i) is called the (n, k) *recovery property*. We say that a coding scheme satisfies the *maximal-distance separable (MDS) property* if the (n, k) recovery property is satisfied and each node stores B/k units of data. The MDS property can be achieved by conventional MDS codes such as the Reed-Solomon (RS) codes. However, the communication and traffic required in repairing a failed node is very large if RS codes are employed, as the whole file must be downloaded before we re-encode the lost data in the failed node. The amount of traffic, measured in the number of packets transmitted from the surviving nodes to the new node, is coined *repair bandwidth* by Dimakis *et al.* in [1]. A lower bound on repair bandwidth is derived in the same work. A coding scheme with repair bandwidth attaining the lower bound is called a *regenerating code*.

The repair of failed storage nodes can be carried out in two ways. In the first one, called *exact repair*, the contents of the new nodes are exactly the same as the failed ones. The second is *functional repair*, in which the content need not be recovered exactly, but the (n, k) recovery property is maintained. Exact repair has the advantage that we can store the data file in an uncoded form in some nodes, called the *systematic nodes*, while the other nodes store the parity-check data. In case we want to look up a small portion of the data file, we can connect to the node which holds that particular portion, without downloading the whole file. There are several existing constructions of regenerating codes for exact repair. One approach is to apply idea from *interference alignment* [2], [3], which is a concept in wireless communication for characterizing the degree of freedom of a wireless network. The regenerating code by Suh and Ramchandran [4] is one class of regenerating code constructed using this technique.

The Suh-Ramchandran code is designed for repairing single failure. For multiple failures, it was shown by Hu *et al.* in [5] that by enabling data exchange, the repair bandwidth per new node can be further reduced. Suppose that we want to repair r failures simultaneously. The repair process is divided into two phases. In the first phase, each newcomer downloads β_1 packets from a set of d surviving nodes. The system parameter d is often called the *repair degree*. In the second phase, each pair of newcomers exchange β_2 packets in both directions. A regenerating code which repairs multiple-node failure jointly according to this two-phase protocol will be referred to as *cooperative* or *collaborative* regenerating code. The repair bandwidth per new node is denoted by $\gamma = d\beta_1 + (r-1)\beta_2$.

It was shown in [6] that for any cooperative regenerating code satisfying the MDS property, the repair bandwidth is lower bounded by

$$\gamma \geq \frac{B(d+r-1)}{k(d+r-k)}. \tag{1}$$

A cooperative regenerating code satisfying the MDS property and (1) with equality is called minimum-storage cooperative regenerating (MSCR) code. On the other hand, if the storage in each node is allowed to be larger than B/k, then the repair bandwidth of a cooperative regenerating code is lower bounded by

$$\gamma \geq \frac{B(2d+r-1)}{k(2d+r-k)}. \tag{2}$$

A cooperative regenerating code satisfying (2) with equality is called minimum-bandwidth cooperative regenerating (MBCR) code. When $r = 1$, the bounds in (1) and (2) reduce to those for single-node repair in [1].

There are some existing constructions of exact-repair MSCR and MBCR codes [7]–[11]. These constructions are summarized in Table I. For functional repair, the tradeoff curve between repair bandwidth and storage is derived in [12].

This work was partially supported by a grant from the University Grants Committee of the Hong Kong Special Administrative Region, China (Project No. AoE/E-02/08).

TABLE I

EXPLICIT CONSTRUCTIONS OF COOPERATIVE REGENERATING CODES.

type	parameters	reference
MSCR	$n \geq d + r,\, k = d$	[7]
MBCR	$n = d + r,\, d = k$	[8]
MBCR	$n = d + r,\, d \geq k,\, r \geq 1$	[9]
MBCR	$n \geq d + r,\, d \geq k,\, r \geq 1$	[10]
MSCR	$n = d + r,\, d \geq k,\, k = 2,\, r = 2$	[11]
MSCR	$n = d + r = 2k,\, k \geq 3,\, r = 2$	this paper

The objective of this paper is to show that the structure of the Suh-Ramchandran regenerating code also supports multiple-node repair. This disproves the assertion in [11] that "it is not possible to repair exactly MSCR code with $k \geq 3$, $r \geq 2, d > k$ in the scalar case." After reviewing the Suh-Ramchandran construction in Section II, we state the main result of this paper in Section III. In Section IV, an example with $(n, k) = (6, 3)$ is given. The proof of the main theorem is stated in Section V.

II. The Suh-Ramchandran Construction

In the Suh-Ramchandran construction, the number of nodes, n, can be any integer larger than or equal to $2k$. For the ease of presentation, we focus on the case $n = 2k$ in this paper. We will use notations different from those in [4], in order to emphasize the symmetry of the code, which will be crucial in the derivation of multiple-node recovery process.

Let $\mathbb{F}_q$ denote a finite field of size q. Each data symbol is regarded as a finite field element, and we will use a symbol as a unit of data. A symbol will also be called a *packet*. The data file is divided into many data chunks, each containing $B = k^2$ symbols. All data chunks are encoded and treated in the same way. Hence, we only need to describe the operations on one data chunk, and without loss of generality, we can assume that the data file consists of exactly k^2 symbols.

The construction requires four non-singular $k \times k$ matrices $\mathbf{U} = [u_{ij}]$, $\mathbf{V} = [v_{ij}]$, $\mathbf{P} = [p_{ij}]$ and $\mathbf{Q} = \mathbf{P}^{-1} = [q_{ij}]$ over $\mathbb{F}_q$, satisfying

$$\mathbf{U} = \mathbf{VP} \text{ and } \mathbf{V} = \mathbf{UQ}. \tag{3}$$

Denote the columns of $\mathbf{U}$ by $\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_k$, and the columns of $\mathbf{V}$ by $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_k$. The columns of $\mathbf{U}$ and $\mathbf{V}$ are regarded as bases of $\mathbb{F}_q^k$, and the matrices $\mathbf{P}$ and $\mathbf{Q}$ are the change-of-basis matrices; the transformations in (3) are equivalent to

$$\mathbf{u}_i = p_{1i}\mathbf{v}_1 + p_{2i}\mathbf{v}_2 + \cdots + p_{ki}\mathbf{v}_k,$$
$$\mathbf{v}_i = q_{1i}\mathbf{u}_1 + q_{2i}\mathbf{u}_2 + \cdots + q_{ki}\mathbf{u}_k,$$

for $i = 1, 2, \ldots, k$. Let

$$\hat{\mathbf{U}} := (\mathbf{U}^t)^{-1} \text{ and } \hat{\mathbf{V}} := (\mathbf{V}^t)^{-1}, \tag{4}$$

where the superscript ^{t} denotes the transpose operator. The columns of $\hat{\mathbf{U}}$ (resp. $\hat{\mathbf{V}}$) form the dual basis of $\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_k$ (resp. $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_k$). Let the columns of $\hat{\mathbf{U}}$ be $\hat{\mathbf{u}}_1, \hat{\mathbf{u}}_2, \ldots, \hat{\mathbf{u}}_k$, and the columns of $\hat{\mathbf{V}}$ be $\hat{\mathbf{v}}_1, \hat{\mathbf{v}}_2, \ldots, \hat{\mathbf{v}}_k$.

Each node stores a column vector of length k over $\mathbb{F}_q$. For $i = 1, 2, \ldots, k$, let the vector stored in node i be denoted by $\mathbf{x}_i$, and the vector stored in node $k+i$ be $\mathbf{y}_i$. Let $\mathbf{X}$ (resp. $\mathbf{Y}$) be the $k \times k$ matrix whose columns are $\mathbf{x}_i$ (resp. $\mathbf{y}_i$).

The Suh-Ramchandran regenerating code can be constructed in two ways. In the first way, the data packets stored in nodes 1 to k are uncoded symbols, and the packets stored in nodes $k+1$ to $2k$ are obtained by some linear transformation on the packets in nodes 1 to k, i.e., nodes 1 to k are the systematic nodes, and nodes $k+1$ to $2k$ are the parity-check nodes. The parity-check symbols in nodes $k+1$ to n are generated by

$$\mathbf{Y} = \delta\hat{\mathbf{V}}\mathbf{X}^t\mathbf{U} + \epsilon\mathbf{XP}. \tag{5}$$

The variable δ and ϵ are elements in $\mathbb{F}_q$ to be determined later. If we let

$$\mathbf{z}_j := \sum_{\ell=1}^{k} p_{\ell j}\mathbf{x}_\ell$$

to be the j-th column in matrix $\mathbf{XP}$, then we write (5) in an alternate way as

$$\mathbf{y}_j = \Big(\delta \sum_{i=1}^{k} \hat{\mathbf{v}}_i \mathbf{u}_j^t \mathbf{x}_i\Big) + \epsilon\mathbf{z}_j, \tag{5'}$$

for $j = 1, 2, \ldots, k$.

In the second way of constructing the Suh-Ramchandran regenerating code, the packets in nodes $k+1$ to $2k$ are treated as information packets, while the packets in nodes 1 to k are parity-check packets. The matrix $\mathbf{X}$ is obtained from $\mathbf{Y}$ by

$$\mathbf{X} = \delta'\hat{\mathbf{U}}\mathbf{Y}^t\mathbf{V} + \epsilon'\mathbf{YQ}, \tag{6}$$

where δ' and ϵ' are elements in $\mathbb{F}_q$. We use the notation

$$\mathbf{z}'_j := \sum_{\ell=1}^{k} q_{\ell j}\mathbf{y}_\ell$$

to denoted the j-th column of matrix $\mathbf{YQ}$. For $j = 1, 2, \ldots, k$, the data stored in node j can be expressed as

$$\mathbf{x}_j = \Big(\delta' \sum_{i=1}^{k} \hat{\mathbf{u}}_i \mathbf{v}_j^t \mathbf{y}_i\Big) + \epsilon'\mathbf{z}'_j. \tag{6'}$$

The equivalence of these two ways of encoding is shown in the next theorem.

Theorem 1. *Let $F(\mathbf{X}) = \delta\hat{\mathbf{V}}\mathbf{X}^t\mathbf{U} + \epsilon\mathbf{XP}$ and $G(\mathbf{Y}) = \delta'\hat{\mathbf{U}}\mathbf{Y}^t\mathbf{V} + \epsilon'\mathbf{YQ}$ be linear transformations from the vector space of $k \times k$ matrices to itself. If we choose δ, δ', ϵ and ϵ' such that*

$$\delta\delta' + \epsilon\epsilon' = 1, \text{ and} \tag{7}$$
$$\epsilon\delta' + \delta\epsilon' = 0, \tag{8}$$

then the compositions $F \circ G$ and $G \circ F$ are the identity transformation.

Proof: For all $k \times k$ matrices $\mathbf{X}$, we have

$$\begin{aligned} G(F(\mathbf{X})) &= \delta'\hat{\mathbf{U}}(\delta\mathbf{U}^t\mathbf{X}\hat{\mathbf{V}}^t + \epsilon\mathbf{P}^t\mathbf{X}^t)\mathbf{V} \\ &\quad + \epsilon'(\delta\hat{\mathbf{V}}\mathbf{X}^t\mathbf{U} + \epsilon\mathbf{XP})\mathbf{Q} \\ &= (\delta\delta' + \epsilon\epsilon')\mathbf{X} + (\epsilon\delta' + \delta\epsilon')\hat{\mathbf{V}}\mathbf{X}^t\mathbf{V} = \mathbf{X}. \end{aligned}$$

The proof of $F(G(\mathbf{Y})) = \mathbf{Y}$ is similar. ■

In [4], Suh and Ramchandran prove the following.

Theorem 2 ([4]). *The Suh-Ramchandran regenerating codes satisfies the MDS property if all square submatrices of matrix* $\mathbf{P}$ *are non-singular.*

We will call a matrix *super-regular* if all square submatrices are non-singular. It can be proved that the inverse of a super-regular matrix is also super-regular. Therefore in Theorem 2, it is equivalent to pick the matrix $\mathbf{Q}$ to be super-regular.

III. Main Result

The main result of this paper is to show that the Suh-Ramchandran regenerating code, which is originally aiming at repairing single-node failure, can repair the following patterns of multiple-node failures with minimal repair bandwidth.

Theorem 3. *Suppose that in the Suh-Ramchandran construction, the parameters* $\mathbf{V}$, $\mathbf{P}$, ϵ, δ, ϵ' *and* δ' *are chosen such that*

- $\mathbf{V}$ *is a* $k \times k$ *non-singular matrices over* $\mathbb{F}_q$,
- $\mathbf{P}$ *is a* $k \times k$ *super-regular matrices over* $\mathbb{F}_q$,
- ϵ, δ, ϵ' *and* δ' *are non-zero and satisfy* (7) *and* (8),
- $p_{ij}q_{ji} \neq 1$ *for all* i *and* j *in* $\{1, 2, \dots, k\}$.

Then we can exactly repair

- r *systematic nodes, for any* r *between 1 and* k,
- r *parity-check nodes, for any* r *between 1 and* k,
- *any pair of systematic node and parity-check node,*

with repair bandwidth attaining the lower bound in (1) *and repair degree* d *equal to* n *minus the number of failed nodes repaired cooperatively.*

The proof of Theorem 3 is given in Section V.

We need to choose the coding coefficients such that the conditions in Theorem 3 are satisfied. First of all, if we square both sides of (7) and (8) and subtract, we get

$$(\delta^2 - \epsilon^2)((\delta')^2 - (\epsilon')^2) = 1.$$

Hence, we have $\delta^2 \neq \epsilon^2$ and $(\delta')^2 \neq (\epsilon')^2$. As the determinant of the 2×2 matrix in

$$\begin{bmatrix} \delta & \epsilon \\ \epsilon & \delta \end{bmatrix} \begin{bmatrix} \delta' \\ \epsilon' \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \tag{9}$$

is necessarily non-zero, we can choose δ and ϵ to be a pair of nonzero elements in $\mathbb{F}_q$ such that $\delta^2 \neq \epsilon^2$, and then obtain ϵ' and δ' by solving (9). The values of ϵ' and δ' so obtained are provably non-zero.

Secondly, for a Cauchy matrix $\mathbf{P} = [(a_i - b_j)^{-1}]$, the (j, i)-entry of $\mathbf{P}^{-1}$ can be calculated by

$$q_{ji} = (a_i - b_j) \frac{\prod_{\ell \neq i}(b_j - a_\ell)}{\prod_{\ell \neq i}(a_i - a_\ell)} \cdot \frac{\prod_{\ell \neq j}(a_i - b_\ell)}{\prod_{\ell \neq j}(b_j - b_\ell)}. \tag{10}$$

See for example [13] for a derivation of (10). Whence, the condition $p_{ij}q_{ji} \neq 1$ is equivalent to

$$\prod_{\ell \neq i}(b_j - a_\ell) \cdot \prod_{\ell \neq j}(a_i - b_\ell) - \prod_{\ell \neq i}(a_i - a_\ell) \cdot \prod_{\ell \neq j}(b_j - b_\ell) \neq 0.$$

Let F_{ij} be the left-hand side of the above equation, regarded as a mutli-variate polynomial in a_i's and b_j's. Constructing a Cauchy matrix $\mathbf{P}$ satisfying the conditions in Theorem 3 amounts to finding a_i's and b_j's such that the product $\prod_{1 \leq i,j \leq k} F_{ij}$ is evaluated to a non-zero constant in $\mathbb{F}_q$. By Schwartz-Zippel lemma (see e.g. [14, Corollary 19.18]), this can be done if the finite field size q is sufficiently large.

Corollary 4. *With sufficiently large finite field* $\mathbb{F}_q$, *we can repair single and double node failures in the Suh-Ramchandran regenerating code with optimal repair bandwidth.*

IV. An Example for $n = 6$ and $k = 3$

In this section, we illustrate how to repair two node failures in the rate-$1/2$ Suh-Ramchandran code for $n = 6$ nodes.

Encoding. There are $B = 9$ symbols to be encoded and distributed to $n = 6$ storage nodes. Let us agree that the first three nodes are systematic nodes, and the last three nodes are parity-check nodes. Each node stores a column vector of length 3. We let $\mathbf{V} = [\mathbf{v}_1|\mathbf{v}_2|\mathbf{v}_3]$ be a non-singular 3×3 matrices, and $\mathbf{P} = [p_{ij}]_{i,j=1}^3$ be a Cauchy matrix, so that the MDS property is guaranteed by Theorem 2. Let $\mathbf{U} = [\mathbf{u}_1|\mathbf{u}_2|\mathbf{u}_3] = \mathbf{VP}$ and denote the inverse of $\mathbf{P}$ by $\mathbf{Q} = \mathbf{P}^{-1} = [q_{ij}]_{i,j=1}^3$.

The encoding is illustrated in the following table:

Node	Content
1	$\mathbf{x}_1$
2	$\mathbf{x}_2$
3	$\mathbf{x}_3$
4	$\mathbf{y}_1 = \delta \sum_{j=1}^3 \hat{\mathbf{v}}_j \mathbf{u}_1^t \mathbf{x}_j + \epsilon \mathbf{z}_1$
5	$\mathbf{y}_2 = \delta \sum_{j=1}^3 \hat{\mathbf{v}}_j \mathbf{u}_2^t \mathbf{x}_j + \epsilon \mathbf{z}_2$
6	$\mathbf{y}_3 = \delta \sum_{j=1}^3 \hat{\mathbf{v}}_j \mathbf{u}_3^t \mathbf{x}_j + \epsilon \mathbf{z}_3$

Repair. Upon the failure of two storage nodes, each surviving node sends a linear combination of the stored symbols to each of the failed node. The first phase of the repair procedure is as follows.

1) If node i is one of the failed node, for $i = 1, 2, 3$, a surviving node takes the inner product of the stored vector and $\mathbf{v}_i$, and sends it to newcomer i.
2) If node $3 + j$ is one of the failed node, for $j = 1, 2, 3$, a surviving node takes the inner product of the stored vector and $\mathbf{u}_j$, and sends it to newcomer $3 + j$.

By the symmetry of the code structure, it is sufficient to discuss the repair of (i) two parity-check nodes, and (ii) one systematic node and one parity check node.

Repair of two parity-check nodes. Without loss of generality, we consider the repair of nodes 4 and 5. After the first phase of the repair process, newcomer 4 receives four symbols,

$$\mathbf{u}_1^t \mathbf{x}_1,\ \mathbf{u}_1^t \mathbf{x}_2,\ \mathbf{u}_1^t \mathbf{x}_3 \text{ and } \mathbf{u}_1^t \mathbf{y}_3 = \delta \mathbf{u}_3^t \mathbf{z}_1 + \epsilon \mathbf{u}_1^t \mathbf{z}_3.$$

The symbols received by newcomer 5 are

$$\mathbf{u}_2^t \mathbf{x}_1,\ \mathbf{u}_2^t \mathbf{x}_2,\ \mathbf{u}_2^t \mathbf{x}_3 \text{ and } \mathbf{u}_2^t \mathbf{y}_3 = \delta \mathbf{u}_3^t \mathbf{z}_2 + \epsilon \mathbf{u}_2^t \mathbf{z}_3.$$

Recall that newcomer 5 wants to compute

$$\mathbf{y}_2 = \delta(\hat{\mathbf{v}}_1\mathbf{u}_2^t\mathbf{x}_1 + \hat{\mathbf{v}}_2\mathbf{u}_2^t\mathbf{x}_2 + \hat{\mathbf{v}}_3\mathbf{u}_2^t\mathbf{x}_3) + \epsilon\mathbf{z}_2. \qquad (11)$$

The first term can be obtained from $\mathbf{u}_2^t\mathbf{x}_1$, $\mathbf{u}_2^t\mathbf{x}_2$ and $\mathbf{u}_2^t\mathbf{x}_3$. For the second term, newcomer 5 first calculates

$$\mathbf{u}_2^t\mathbf{z}_2 = p_{12}\mathbf{u}_2^t\mathbf{x}_1 + p_{22}\mathbf{u}_2^t\mathbf{x}_2 + p_{32}\mathbf{u}_2^t\mathbf{x}_3,$$
$$\mathbf{u}_3^t\mathbf{z}_2 = \frac{1}{\delta}\Big(\mathbf{u}_2^t\mathbf{y}_3 - \epsilon p_{13}\mathbf{u}_2^t\mathbf{x}_1 - \epsilon p_{23}\mathbf{u}_2^t\mathbf{x}_2 - \epsilon p_{33}\mathbf{u}_2^t\mathbf{x}_3\Big).$$

and then asks newcomer 4 for a copy

$$\mathbf{u}_1^t\mathbf{z}_2 = p_{11}\mathbf{u}_1^t\mathbf{x}_1 + p_{21}\mathbf{u}_1^t\mathbf{x}_2 + p_{31}\mathbf{u}_1^t\mathbf{x}_3,$$

which can be computed by newcomer 4. In the computation of $\mathbf{u}_3^t\mathbf{z}_2$, it is obvious that we need to impose the condition that $\delta \neq 0$. Then, by the linear independence of $\mathbf{u}_1$, $\mathbf{u}_2$ and $\mathbf{u}_3$, newcomer 5 can regenerating the second term in (11).

Similarly, newcomer 4 can regenerate $\mathbf{y}_1$ after newcomer 5 has sent $\mathbf{u}_2^t\mathbf{z}_1$ to newcomer 4.

Repair of a systematic node and a parity-check node. Without loss of generality, we consider the repair of nodes 1 and 5. After the first phase of the repair process, newcomer 1 receives $\mathbf{v}_1^t\mathbf{x}_2$, $\mathbf{v}_1^t\mathbf{x}_3$,

$$\mathbf{v}_1^t\mathbf{y}_1 = \delta\mathbf{u}_1^t\mathbf{x}_1 + \epsilon\mathbf{v}_1^t\mathbf{z}_1, \text{ and } \mathbf{v}_1^t\mathbf{y}_3 = \delta\mathbf{u}_3^t\mathbf{x}_1 + \epsilon\mathbf{v}_1^t\mathbf{z}_3,$$

while newcomer 5 receives $\mathbf{u}_2^t\mathbf{x}_2$, $\mathbf{u}_2^t\mathbf{x}_3$,

$$\mathbf{u}_2^t\mathbf{y}_1 = \delta\mathbf{u}_1^t\mathbf{z}_2 + \epsilon\mathbf{u}_2^t\mathbf{z}_1, \text{ and } \mathbf{u}_2^t\mathbf{y}_3 = \delta\mathbf{u}_3^t\mathbf{z}_2 + \epsilon\mathbf{u}_2^t\mathbf{z}_3.$$

Newcomer 5 computes a linear combination of the received symbols,

$$q_{11}\mathbf{u}_2^t\mathbf{y}_1 + q_{31}\mathbf{u}_2^t\mathbf{y}_3 + (\delta+\epsilon)[p_{22}q_{21}\mathbf{u}_2^t\mathbf{x}_2 + p_{32}q_{21}\mathbf{u}_2^t\mathbf{x}_3].$$

The coefficients are chosen so that it can be simplified to

$$\delta\mathbf{v}_1^t\mathbf{z}_2 + (\epsilon - (\epsilon+\delta)p_{12}q_{21})\mathbf{u}_2^t\mathbf{x}_1, \qquad (12)$$

which is a linear combination of $\mathbf{v}_1^t\mathbf{z}_2$ and $\mathbf{u}_2^t\mathbf{x}_1$. (We have used the orthogonality relation $\sum_\ell p_{i\ell}q_{\ell j}$ is equal to the Kronecker delta function δ_{ij}.) In the second phase of the repair process, newcomer 5 sends the symbol in (12) to newcomer 1.

Since newcomer 1 knows $\mathbf{v}_1^t\mathbf{x}_2$ and $\mathbf{v}_1^t\mathbf{x}_3$, newcomer 1 can compute

$$\big(\delta p_{12}\mathbf{v}_1^t + (\epsilon - (\epsilon+\delta)p_{12}q_{21})\mathbf{u}_2^t\big)\mathbf{x}_1$$

by subtracting $\delta p_{22}\mathbf{v}_1^t\mathbf{x}_2$ and $\delta p_{32}\mathbf{v}_1^t\mathbf{x}_3$. Next, newcomer 1 calculates

$$\mathbf{v}_1^t\mathbf{y}_1 - \epsilon p_{21}\mathbf{v}_1^t\mathbf{x}_2 - \epsilon p_{31}\mathbf{v}_1^t\mathbf{x}_3 = (\delta\mathbf{u}_1^t + \epsilon p_{11}\mathbf{v}_1^t)\mathbf{x}_1, \text{ and}$$
$$\mathbf{v}_1^t\mathbf{y}_3 - \epsilon p_{23}\mathbf{v}_1^t\mathbf{x}_2 - \epsilon p_{33}\mathbf{v}_1^t\mathbf{x}_3 = (\delta\mathbf{u}_3^t + \epsilon p_{13}\mathbf{v}_1^t)\mathbf{x}_1.$$

The vector $\mathbf{x}_1$ can be recovered if the matrix

$$\begin{bmatrix} (\epsilon - (\epsilon+\delta)p_{12}q_{21})\mathbf{u}_2^t + \delta p_{12}\mathbf{v}_1^t \\ \delta\mathbf{u}_1^t + \epsilon p_{11}\mathbf{v}_1^t \\ \delta\mathbf{u}_3^t + \epsilon p_{13}\mathbf{v}_1^t \end{bmatrix}$$

is non-singular.

Using the symmetry of the code, newcomer 5 can recover the lost information in a similar way.

V. Proof of the Main Theorem

We use the first encoding method of the Suh-Ramchandran code; the entries in $\mathbf{X}$ are the source symbols and the entries in $\mathbf{Y}$ are the parity-check symbols calculated by (5). In the first phase of the repair procedure, the packet sent from a surviving node to a newcomer is computed as follows:

1) If node i is one of the failed node, for $i = 1, 2, \dots, k$, then a surviving node takes the inner product of the stored vector and $\mathbf{v}_i$, and sends it to newcomer i.
2) If node $k+i$ is one of the failed node, for $i = 1, 2, \dots, k$, then a surviving node takes the inner product of the stored vector and $\mathbf{u}_i$, and sends it to newcomer $k+i$.

Repair of r parity-check or systematic nodes, $1 \leq r \leq k$**.** By the symmetry between $\mathbf{X}$ and $\mathbf{Y}$, it suffices to consider the repair of parity-check nodes.

Suppose that nodes $k+1$ to $k+r$ fail. For $i = 1, 2, \dots, r$, the symbols received by newcomer $k+i$ are $\mathbf{u}_i^t\mathbf{x}_1$ to $\mathbf{u}_i^t\mathbf{x}_k$ and $\mathbf{u}_i^t\mathbf{y}_j = \delta\mathbf{u}_j^t\mathbf{z}_i + \epsilon\mathbf{u}_i^t\mathbf{z}_j$ for $j = r+1, r+2, \dots, k$.

Recall that newcomer $k+i$ wants to regenerate

$$\Big(\delta\sum_{\ell=1}^{k}\hat{\mathbf{v}}_\ell\mathbf{u}_i^t\mathbf{x}_\ell\Big) + \epsilon\mathbf{z}_i.$$

The first term is known to newcomer $k+i$ after the first phase, and can be reconstructed from $\mathbf{u}_i^t\mathbf{x}_1$ to $\mathbf{u}_i^t\mathbf{x}_k$. For the second term, newcomer $k+i$ calculates

$$\mathbf{u}_i^t\mathbf{z}_i = \sum_{\ell=1}^{k} p_{\ell i}\mathbf{u}_i^t\mathbf{x}_\ell, \quad \text{and}$$
$$\mathbf{u}_j^t\mathbf{z}_i = \frac{1}{\delta}\mathbf{u}_i^t\mathbf{y}_j - \frac{\epsilon}{\delta}\Big(\sum_{\ell=1}^{k} p_{\ell j}\mathbf{u}_i^t\mathbf{x}_\ell\Big),$$

for $j = r+1, r+2, \dots, k$, and asks the other $r-1$ newcomers each for a copy of $\mathbf{u}_j^t\mathbf{z}_i$, for $1 \leq j \leq r, j \neq i$. Using the fact that $\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_k$ are linearly independent, newcomer $k+i$ can then solve for $\mathbf{z}_i$.

Repair of a systematic node and a parity-check node. Suppose nodes a and $k+b$ fail, where a and b are integers between 1 and k. We want to replace them by newcomer a and newcomer $k+b$. Let $[k]$ denote $\{1, 2, \dots, k\}$.

After the first phase of the repair process, newcomer a receives

$$\mathbf{v}_a^t\mathbf{x}_i \text{ for } i \in [k] \setminus \{a\}, \text{ and}$$
$$\mathbf{v}_a^t\mathbf{y}_j = \delta\mathbf{u}_j^t\mathbf{x}_a + \epsilon\mathbf{v}_a^t\mathbf{z}_j \text{ for } j \in [k] \setminus \{b\},$$

and newcomer $k+b$ receives

$$\mathbf{u}_b^t\mathbf{x}_i \text{ for } i \in [k] \setminus \{a\}, \text{ and}$$
$$\mathbf{u}_b^t\mathbf{y}_j = \delta\mathbf{u}_j^t\mathbf{z}_b + \epsilon\mathbf{u}_b^t\mathbf{z}_j \text{ for } j \in [k] \setminus \{b\}.$$

In the second phase, newcomer $k+b$ sends the linear combination

$$\sum_{j\neq b} q_{ja}\mathbf{u}_b^t\mathbf{y}_j + (\delta+\epsilon)\sum_{i\neq a} p_{ib}q_{ba}\mathbf{u}_b^t\mathbf{x}_i$$
$$= \delta\mathbf{v}_a^t\mathbf{z}_b + (\epsilon - (\epsilon+\delta)p_{ab}q_{ba})\mathbf{u}_b^t\mathbf{x}_a,$$

where j runs over $[k] \setminus \{b\}$ and i runs over $[k] \setminus \{a\}$, to newcomer a. Newcomer a then calculates

$$\big(\delta p_{ab}\mathbf{v}_a^t + (\epsilon - (\epsilon+\delta)p_{ab}q_{ba}\mathbf{u}_b^t\big)\mathbf{x}_a,$$

and $(\delta\mathbf{u}_j^t + \epsilon p_{aj}\mathbf{v}_a^t)\mathbf{x}_a$ for $j \in [k]\setminus\{b\}$. The vector $\mathbf{x}_a$ can be recovered if the following $k \times k$ matrix

$$\begin{bmatrix} (\epsilon - (\epsilon+\delta)p_{ab}q_{ba})\mathbf{u}_b^t + \delta p_{ab}\mathbf{v}_a^t \\ \delta\mathbf{u}_1^t + \epsilon p_{a1}\mathbf{v}_a^t \\ \vdots \\ \delta\mathbf{u}_{b-1}^t + \epsilon p_{a,b-1}\mathbf{v}_a^t \\ \delta\mathbf{u}_{b+1}^t + \epsilon p_{a,b+1}\mathbf{v}_a^t \\ \vdots \\ \delta\mathbf{u}_k^t + \epsilon p_{ak}\mathbf{v}_a^t \end{bmatrix} \tag{13}$$

is non-singular. We will show in Prop. 5 that the determinant of this matrix is non-zero if $p_{ab}q_{ba} \neq 1$.

Using the symmetric of the code, newcomer $k+b$ can recover the lost information after receiving

$$\delta'\mathbf{u}_b^t\mathbf{z}'_a + (\epsilon' - (\epsilon'+\delta')q_{ba}p_{ab})\mathbf{v}_a^t\mathbf{y}_b$$

from newcomer a, provided that $p_{ab}q_{ba} \neq 1$.

Proposition 5. *Suppose that* $\mathbf{V}$, $\mathbf{P}$, ϵ, δ, ϵ' *and* δ' *satisfy the criteria in Theorem 3. Then the determinant of the matrix in* (13) *is non-zero.*

Proof: We divide the proof into two cases.

Case 1: $\epsilon - (\epsilon+\delta)p_{ab}q_{ba} = 0$. In this case, we can row-reduce the matrix in (13) to

$$\begin{bmatrix} \delta p_{ab}\mathbf{v}_a^t \\ \delta\mathbf{U}_{[k]\setminus\{b\}}^t \end{bmatrix} = \begin{bmatrix} \delta p_{ab}\sum_{\ell=1}^k q_{\ell a}\mathbf{u}_\ell^t \\ \delta\mathbf{U}_{[k]\setminus\{b\}}^t \end{bmatrix}$$

where $\mathbf{U}_{[k]\setminus\{b\}}$ denotes the $k \times (k-1)$ matrix

$$\mathbf{U}_{[k]\setminus\{b\}} = \begin{bmatrix} \mathbf{u}_1 & \dots & \mathbf{u}_{b-1} & \mathbf{u}_{b+1} & \dots & \mathbf{u}_k \end{bmatrix}.$$

It can further be row-reduced to a non-singular matrix, and thus has non-zero determinant.

Case 2: $\epsilon - (\epsilon+\delta)p_{ab}q_{ba} \neq 0$. After substituting $\mathbf{v}_a$ by $\sum_{\ell=1}^k q_{\ell a}\mathbf{u}_\ell$, the matrix in (13) can be factored as

$$\begin{bmatrix} \epsilon - \epsilon p_{ab}q_{ba} & \delta p_{ab}\mathbf{q}_{[k]\setminus\{b\},a}^t \\ \epsilon q_{ba}\mathbf{p}_{a,[k]\setminus\{b\}} & \delta\mathbf{I} + \epsilon\mathbf{p}_{a,[k]\setminus\{b\}}\mathbf{q}_{[k]\setminus\{b\},a}^t \end{bmatrix} \begin{bmatrix} \mathbf{u}_b^t \\ \mathbf{U}_{[k]\setminus\{b\}}^t \end{bmatrix} \tag{14}$$

where $\mathbf{I}$ is the $(k-1)\times(k-1)$ identity matrix, $\mathbf{p}_{a,[k]\setminus\{b\}}$ is the column vector

$$\mathbf{p}_{a,[k]\setminus\{b\}} := [p_{a1} \ \cdots p_{a,b-1} \ p_{a,b+1} \cdots p_{ak}]^t,$$

and $\mathbf{q}_{[k]\setminus\{b\},a}$ is the column vector

$$\mathbf{q}_{[k]\setminus\{b\},a} := [q_{1a} \ \cdots q_{b-1,a} \ q_{b+1,a} \cdots q_{ka}]^t.$$

The non-singularity of (13) is equivalent to the non-singularity of the first factor in (14), which in turn can be decomposed as

$$\mathbf{A} + \mathbf{g}\mathbf{h}^t,$$

where

$$\mathbf{A} = \begin{bmatrix} \epsilon - (\epsilon+\delta)p_{ab}q_{ba} & \mathbf{0} \\ \mathbf{0} & \delta\mathbf{I} \end{bmatrix}$$

is a diagonal matrix, and

$$\mathbf{g} = \begin{bmatrix} \delta p_{ab} \\ \epsilon\mathbf{p}_{a,[k]\setminus\{b\}} \end{bmatrix}, \ \mathbf{h} = \begin{bmatrix} q_{ba} \\ \mathbf{q}_{[k]\setminus\{b\},a} \end{bmatrix}$$

are column vectors.

The first summand is non-singular because $\epsilon - (\epsilon+\delta)p_{ab}q_{ba}$ and δ are non-zero. By the Sherman-Morrison formula [15, p.18], we see that the matrix in (13) is invertible if

$$1 + \mathbf{h}^t\mathbf{A}^{-1}\mathbf{g}$$

is non-zero. Using the identity $\sum_{\ell=1}^k p_{a\ell}q_{\ell a} = 1$, the above expression can be simplified to

$$\frac{\epsilon(\epsilon+\delta)(1-p_{ab}q_{ba})^2}{\epsilon - (\epsilon+\delta)p_{ab}q_{ba}},$$

which is nonzero because $\epsilon \neq 0$, $\delta^2 \neq \epsilon^2$, and $p_{ab}q_{ba} \neq 1$. ■

VI. Concluding Remarks

In this paper we show that with the regenerating code constructed by Suh and Ramchandran, which is originally designed for repairing any single node failure, multiple-node failures can also be repaired cooperatively with optimal repair bandwidth. Indeed, we can repair any set of systematic nodes, any set of parity-check nodes, or any pair of nodes. However, the technique that we used in this paper cannot be extended to the optimal repair of one systematic node and two parity-check nodes.

References

[1] A. G. Dimakis, P. B. Godfrey, M. J. Wainwright, and K. Ramchandran, "Network coding for distributed storage system," in *Proc. IEEE Int. Conf. on Computer Comm. (INFOCOM)*, Anchorage, Alaska, May 2007, pp. 2000–2008.

[2] V. R. Cadambe and C. Jafar, "Interference alignment and degrees of freedom of the K-user interference channel," *IEEE Trans. Inf. Theory*, vol. 54, no. 8, pp. 3425–3441, Aug. 2008.

[3] M. A. Maddah-Ali, A. S. Motahari, and A. K. Khandani, "Communication over MIMO X channels: Interference alignment, decomposition, and performance analysis," *IEEE Trans. Inf. Theory*, vol. 54, no. 8, pp. 3457–3470, Aug. 2008.

[4] C. Suh and K. Ramchandran, "Exact-repair MDS code construction using interference alignment," *IEEE Trans. Inf. Theory*, vol. 57, no. 3, pp. 1425–1442, Mar. 2011.

[5] Y. Hu, Y. Xu, X. Wang, C. Zhan, and P. Li, "Cooperative recovery of distributed storage systems from multiple losses with network coding," *IEEE J. on Selected Areas in Commun.*, vol. 28, no. 2, pp. 268–276, Feb. 2010.

[6] A.-M. Kermarrec, N. Le Scouarnec, and G. Straub, "Repairing multiple failures with coordinated and adaptive regenerating codes," in *Proc. Int. Symp. on Network Coding (Netcod)*, Beijing, Jul. 2011, pp. 88–93.

[7] K. W. Shum, "Cooperative regenerating codes for distributed storage systems," in *IEEE Int. Conf. Comm. (ICC)*, Kyoto, Jun. 2011, pp. 1–5.

[8] K. W. Shum and Y. Hu, "Exact minimum-repair-bandwidth cooperative regenerating codes for distributed storage systems," in *Proc. IEEE Int. Symp. Inf. Theory*, St. Petersburg, Aug. 2011, pp. 1374–1378.

[9] S. Jiekak and N. Le Scouarnec, "CROSS-MBCR: Exact minimum bandwidth coordinated regenerating codes," arXiv:1207.0854v1 [cs.IT], Jul. 2012.

[10] A. Wang and Z. Zhang, “Exact cooperative regenerating codes with minimum-repair-bandwidth for distributed storage,” arXiv:1207.0879v1 [cs.IT], Jul. 2012.

[11] N. Le Scouarnec, “Exact scalar minimum storage coordinated regenerating codes,” in *Proc. IEEE Int. Symp. Inf. Theory*, Cambridge, Jul. 2012, pp. 1197–1201.

[12] K. W. Shum and Y. Hu, “Cooperative regenerating codes,” 2013, to appear in IEEE Trans. Inf. Theory.

[13] S. Schechter, “On the inversion of certain matrices,” *Mathematical Tables and other Aids to Computation*, vol. 13, no. 66, pp. 73–77, 1956.

[14] R. W. Yeung, *Information theory and network coding*. New York: Springer, 2008.

[15] R. A. Horn and C. R. Johnson, *Matrix analysis*. Cambrdige: Cambridge University Press, 1985.