PAPER *Special Section on Signal Design and Its Applications in Communications*

Some Constructions for Fractional Repetition Codes with Locality 2*

Mi-Young NAM[†a)], **Jung-Hyun KIM**[†b)], ***Nonmembers, and*** **Hong-Yeop SONG**[†c)], ***Member***

SUMMARY In this paper, we examine the locality property of the original Fractional Repetition(FR) codes and propose two constructions for FR codes with better locality. For this, we first derive the capacity of the FR codes with locality 2, that is the maximum size of the file that can be stored.

Construction 1 generates an FR code with repetition degree 2 and locality 2. This code is optimal in the sense of achieving the capacity we derived.

Construction 2 generates an FR code with repetition degree 3 and locality 2 based on 4-regular graphs with girth g. This code is also optimal in the same sense.

key words: *Regenerating Codes, Fractional Repetition Codes, Locality, Locally Repairable Codes*

1. Introduction

Distributed storage systems (DSS) store huge amount of data across the network of nodes, which are geographically separated, to avoid the permanent loss of data due to a disk failure. To provide reliability, redundancy must be introduced.

Dimakis et al. found that there is a fundamental trade-off between storage space and repair bandwidth [1] [2]. And they introduced a *regenerating code* which is a class of codes that achieve every point on the trade-off curve. Among these points, two extreme points named as minimum storage regime (MSR) and the minimum bandwidth regime (MBR), are of special interest. Several explicit constructions for this regime have been proposed [6–9].

A code is said to have *locality* r if every node can be repaired from accessing at most r other nodes [3]. An erasure code with locality r is called as a locally repairable code (LRC). An LRC with locality r is said to have availability t if every symbol has t disjoint *repair sets* of size at most r [4] [5].

Repair-by-transfer refers to the repair process with only transfer of stored data without any arithmetic operations [7]. A repetition code can carry out the repair-by-transfer. Also a serially concatenated code using a repetition code as its inner code inherits the efficient repair process of the repetition code. The minimum distance of the concatenated code is at least $d_1 d_2$ where d_1 and d_2 is the minimum distance of each component code [18]. We can improve the reliability of the concatenated code when we use an MDS code as an outer code. Fractional repetition(FR) codes are obtained from this concatenation framework [9] [10].

The repair process implied in the FR codes is optimal in terms of the repair-bandwidth and the computational complexity. When a single node of size α fails, the node can be repaired by contacting other α nodes, hence, the locality must be α. Therefore, the locality increases as the number α of symbols per node increases. In this paper, we modify the FR code so that it has a fixed locality 2.

In the remaining of the paper, we first give a brief summary of fractional repetition codes in section 2. In section 3, we derive the upper bound for the size of the file that can be stored by using FR codes with locality 2. Two explicit constructions for FR codes with locality 2 that could store the file with the maximum possible size are proposed. One construction is for the code with only 1 repair set. And the other generates codes with 2 disjoint repair sets. In section 4, we conclude the paper.

2. Preliminary Discussion

2.1 Notations

We define notations used in this paper as follows.

- M: the size of the file.
- θ: the length of the MDS codeword.
- K: reconstruction degree which is the number of nodes that are contacted by a data collector.
- n: the number of storage nodes in the system.
- α: the number of symbols per each node, also called as the storage size.
- ρ: symbol repetition degree.
- $\Omega = \{0, 1, ..., \theta - 1\}$: a set of symbol positions of the MDS codeword of length θ.

[†]The authors are with the Yonsei University

[*]This research was supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education(No. 2013R1A1A2062061). This paper is a full version of 2015 IWSDA proceeding paper "Locally Repairable Fractional Repetition Codes".

a) E-mail: my.nam@yonsei.ac.kr
b) E-mail: jh.kim06@yonsei.ac.kr
c) E-mail: hysong@yonsei.ac.kr

DOI: 10.1587/transfun.E98.A.1

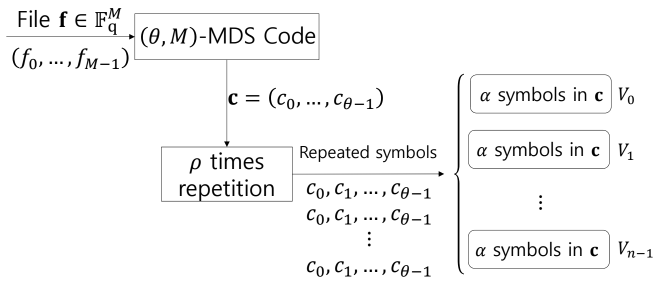


Fig. 1 Encoding of FR codes

- V_i: an α-subset of symbols of Ω that is stored in the i-th storage node, for $i = 0, 1, ..., n-1$.

2.2 Review of the original FR codes

This subsection is a brief summary of the original FR codes from [9]. A DSS consists of n storage nodes, each of which stores α symbols. The symbol is an element of a finite field $\mathbb{F}_q$. A user, also called as a data collector, must be able to reconstruct its stored file of size M by contacting any K out of n nodes, for given $K \leq n$. When a node fails, the system is repaired by replacing the failed node with a *newcomer node*. The newcomer node contacts d surviving nodes and downloads β symbols from each. By using the downloaded symbols, the newcomer node regenerates the symbols that could carry out the same role as the symbols which have been stored in the failed node. In the following, we use an n-set for a set of size n, and an α-subset for a subset of size α.

Definition 1: [9] Given a (θ, M) MDS code and the set of symbol indices $\Omega = \{0, 1, \dots, \theta-1\}$, consider a collection of n α-subsets $V_0, \dots, V_{n-1}$ of Ω. This collection of n α-subsets is called a $((\theta, M), K, (n, \alpha, \rho))$ FR code $\mathcal{C}_{\mathrm{FR}}$ if each symbol of Ω belongs to exactly ρ α-subsets in $\mathcal{C}_{\mathrm{FR}}$.

Figure 1 shows the encoding of an FR code consisting of two steps: (1) MDS encoding of data file of size M into a codeword of length θ, (2) repetition of each symbol ρ times and distribution of them into n storage nodes each of size α. Therefore, ρ is called the repetition degree of $\mathcal{C}_{\mathrm{FR}}$, and it must satisfy

$$\rho\theta = n\alpha. \tag{1}$$

An FR code is said to be *universally good* if it satisfies [9]:

$$|V_i \cap V_j| \leq 1, \tag{2}$$

for $V_i, V_j \in \mathcal{C}_{\mathrm{FR}}, i \neq j \in \{0, 1, ..., n-1\}$. Several explicit constructions for some universally good FR codes have been proposed [11–16]. In fact, most of these constructions are closely related to some combinatorial design problems.

We note that K is called the reconstruction degree of $\mathcal{C}_{\mathrm{FR}}$ because of the following. A data collector contacting any K nodes out of n storage nodes must be able to download at least M coded symbols. These can be used for erasure decoding of the MDS codeword of length θ, from which the original file of size M can be reconstructed. Let $M(K)$ denote the maximum file size of an FR code with given parameters.

Definition 2: [9] The FR capacity $\mathcal{M}(K)$ is the upper bound on the maximum file size $M(K)$ of a $((\theta, M), K, (n, \alpha, \rho))$ FR code. Then the capacity $\mathcal{M}(K)$ is defined as

$$\mathcal{M}(K) \triangleq \max_{\mathcal{C}} \left(\min_{S} \Big| \bigcup_{i \in S} V_i \Big| \right), \tag{3}$$

where $\mathcal{C} = \{V_0, V_1, ..., V_{n-1}\}$ is a $((\theta, M), K, (n, \alpha, \rho))$ FR code and S is a K-subset of $\{0, 1, ..., n-1\}$.

The upper bound on the capacity of FR codes is presented in [16]:

$$\mathcal{M}(K) \leq \varphi(K), \tag{4}$$

where $\varphi(1) = \alpha$, $\varphi(K+1) = \varphi(K) + \alpha - \left\lceil \frac{\rho\varphi(K) - K\alpha}{n-K} \right\rceil$.

An FR code is said to be *optimal* if it can store a file of size which is equal to the capacity [16].

Remark 1: There are several different explicit constructions of FR codes with the same parameters $n, \alpha, \rho, \theta, K$. The maximum file size of a code $\mathcal{C} = \{V_0, V_1, ..., V_{n-1}\}$ is the smallest number of distinct symbols from *any* K storage nodes. The capacity of $((\theta, M), K, (n, \alpha, \rho))$ FR codes is the maximum file size of a code $\mathcal{C}$, where $\mathcal{C}$ is an FR code which maximizes $\left(\min_S \left| \bigcup_{i \in S} V_i \right| \right)$ among all FR codes with the same parameters.

2.3 Locality of FR codes

In this subsection, we examine the FR codes in terms of locality. The locality of the universally good FR code is the same as α. A code in Fig. 2 is a $((6, M), K, (4, 3, 2))$ universally good FR code.

If an FR code has the minimum locality $r = 1$, every storage node has at least one node storing the same α symbols. Figure 3 shows a $((6, M), K, (4, 3, 2))$ FR code with locality 1. Both codes can repair all the single node failures *locally.*

The following simple comparison shows the additional benefit of small locality. If two nodes fail simultaneously, an FR code with locality 1 can repair some of them locally. If V_0 and V_2 fail simultaneously, then an FR code with locality 1 in Fig. 3 can repair them locally. A universally good FR code in Fig. 2 cannot repair any of them locally.

In a code with locality r, if a node i and one of the other $(n-r-1)$ nodes which are not in the repair

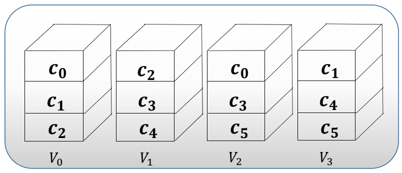


Fig. 2 A universally good $((6, M), K, (4, 3, 2))$ FR code

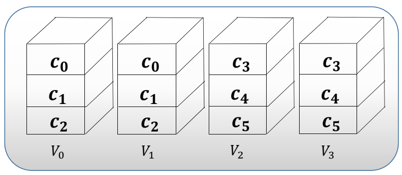


Fig. 3 A $((6, M), K, (4, 3, 2))$ FR code with locality 1

set of the node i fail simultaneously, these two nodes can be repaired locally. Therefore, a code with small locality can repair multiple node failures locally with higher probability than a code with large locality.

The maximum file size of an FR code in Fig. 2 is $M(1) = 3, M(2) = 5, M(3) = 6, M(4) = 6$. The maximum file size of an FR code with locality 1 in Fig. 3 is $M(1) = 3, M(2) = 3, M(3) = 6, M(4) = 6$. The FR codes with locality $r < \alpha$ is some subset of FR codes. Therefore, for given parameters, the capacity of FR codes with locality less than α would be the same or less than the capacity of FR codes. The FR codes with locality $r = 1$ can be constructed straightforward and its capacity is $\mathcal{M}(K) = \left\lceil \frac{K}{\rho} \right\rceil \alpha$, when $\rho = 2, 3$. FR codes with high repetition degree such as $\rho > 3$ is not suitable in regards to practical use. Therefore, we consider the FR codes with locality 2 that is the best locality except for the trivial case of 1 when $\rho \leq 3$.

3. Main Results

3.1 Capacity of FR codes with locality 2

In the following theorem, the upper bound on the capacity of FR codes with locality 2 is derived. It does not consider the parameter n and θ. It is a function of two parameters K and α. Therefore, we use $\mathcal{M}(K, \alpha)$ to denote the capacity.

Theorem 1: Let $\mathcal{C}$ be a $((\theta, M), K, (n, \alpha, \rho))$ FR code with locality 2. Then the capacity $\mathcal{M}(K, \alpha)$ of $\mathcal{C}$ satisfies

$$\mathcal{M}(K, \alpha) \leq \alpha + \sum_{i=0}^{K-2} (\alpha - \beta_{i \pmod 2}), \tag{5}$$

where β_0, β_1 are positive integers such that $\beta_0 + \beta_1 = \alpha$ and $\beta_0 \geq \beta_1 \geq 1$. Furthermore, the right-hand side of (5) is maximized when

$$\beta_0^* \triangleq \lceil \frac{\alpha}{2} \rceil \text{ and } \beta_1^* \triangleq \alpha - \beta_0^*, \tag{6}$$

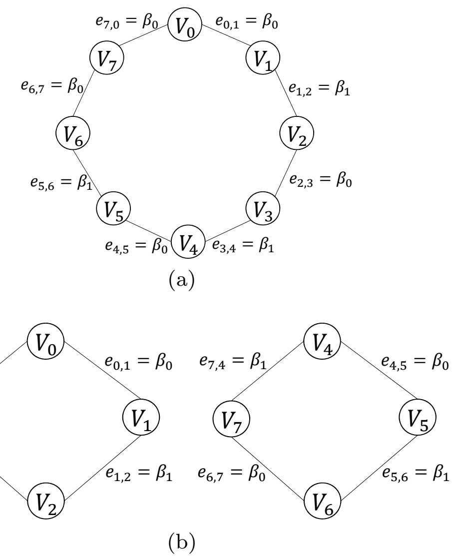


Fig. 4 Graph representations of FR codes with locality 2 and $\rho = 2$. (a) A cycle graph, (b) Two disjoint cycle graphs.

proof:

In an FR code with locality 2 and $\rho \geq 2$, every storage node i storing an α-subset V_i must be repaired by contacting two nodes. β_0 symbols of V_i should be downloaded from some node j, $j \in \{0, 1, ..., n-1\} \backslash \{i\}$, and the remaining $\beta_1 = \alpha - \beta_0$ symbols of V_i should be downloaded from another storage node k, $k \in \{0, 1, ..., n-1\} \backslash \{i, j\}$. That is, for every $i \in \{0, 1, ..., n-1\}$, there must exist a repair set $\{j, k\}$ such that

$$|V_i \cap V_j| = \beta_0, \ |V_i \cap V_k| = \beta_1, \text{ and } V_i \subset (V_j \cup V_k). \tag{7}$$

We can use a graph representation for an FR code. Let $\mathcal{G}$ be a graph of an FR code with locality 2. In a graph $\mathcal{G} = (V, E)$, every vertex $i \in V$ represents each storage node $i \in \{0, 1, ..., n-1\}$. And an edge (i, j) represents that the intersection of two α-subsets V_i and V_j that are stored in a storage node i and j, respectively. If $V_i \cap V_j \neq \phi$, then an edge $(i, j) \in E$. Every edge $(i, j) \in E$ is labeled with an integer, denoted by $e_{i,j}$, which represents the number of intersecting symbols between V_i and V_j, $e_{i,j} = |V_i \cap V_j| = e_{j,i}$.

At first, consider the FR codes with locality 2 and $\rho = 2$. The graph representing an FR code with locality 2 and $\rho = 2$ would be either a cycle graph or a set of multiple disjoint cycle graphs as shown in Fig. 4. Every vertex is incident with two edges, one is labeled with β_0 and the other is labeled with β_1. Note that, this is forced by the condition that $\rho = 2$ and locality 2.

The capacity $\mathcal{M}(K, \alpha)$ can be obtained by using the graph representation as follows:

$$\mathcal{M}(K, \alpha) = \max_{\mathcal{C}} \min_{S} (K\alpha - \sum_{\substack{i<j \\ i,j \in S}} e_{i,j}), \tag{8}$$

where S is a K-subset of $\{0, 1, ..., n-1\}$. The last term $\left(\sum_{\substack{i<j\\ i,j\in S}} e_{i,j}\right)$ in (8) is the sum of all the $e_{i,j}$ for $i < j \in S$ exactly once. A code $\mathcal{C}$ determines the maximum number of edges that are included in a subgraph induced by K vertices of $\mathcal{G}$. If a graph $\mathcal{G}$ of the code $\mathcal{C}$ has cycles of length less than or equal to K, then in some cases, K edges can be included in a subgraph induced by K vertices of $\mathcal{G}$. On the other hand, if a graph $\mathcal{G}$ has no cycles of length less than or equal to K, then the maximum number of edges in a subgraph induced by K vertices must be $K-1$. Therefore, the code $\mathcal{C}$ which maximizes the value of $\left(\min_S(K\alpha - \sum_{\substack{i<j\\ i,j\in S}} e_{i,j})\right)$ is the code which has no cycle of length less than or equal to K.

Then, for given code $\mathcal{C}$, the K-subset S which minimizes the value of $(K\alpha - \sum_{\substack{i<j\\ i,j\in S}} e_{i,j})$ should be a path that starts from some vertex i_s and ends at the other vertex i_e. The path that maximizes $(\sum_{\substack{i<j\\ i,j\in S}} e_{i,j})$ must have as many edges that are labeled with β_0 as possible.

Therefore, the maximum file size of the FR codes with locality 2 and $\rho = 2$ is

$$\mathcal{M}(K,\alpha) = \alpha + \sum_{i=0}^{K-2}(\alpha - \beta_{i \pmod 2}).$$

Now, consider the FR codes with locality 2 and $\rho \geq 3$. Since the code $\mathcal{C}$ has locality 2, for every storage node i, there also must exist a repair set $\{j,k\}$ which satisfies (7). Then there always exists a set S' of K nodes, the cardinality of the union of which is the same as $M'(K) = \alpha + \sum_{i=0}^{K-2}(\alpha - \beta_{i \pmod 2})$. Then the capacity $\mathcal{M}(K,\alpha)$ of FR codes with locality 2 and $\rho \geq 3$ can be upper-bounded by $M'(K)$.

$$\mathcal{M}(K,\alpha) \leq M'(K,\alpha).$$

Therefore, in both cases of $\rho = 2$ and $\rho \geq 3$, the capacity is upper bounded as (5). □

The $((\theta,M),K,(n,\alpha,\rho))$ FR code with locality 2 is said to be *optimal* if it attains the capacity (5) with equality. In subsections 3.2 and 3.3, we propose explicit constructions for optimal $((\theta,M),K,(n,\alpha,\rho))$ FR codes with locality 2 when $\rho = 2$ and $\rho = 3$, respectively.

3.2 Construction for FR codes with locality 2 when $\rho = 2$

We will give an explicit construction for FR codes with locality 2 for general values of α under the symbol repetition degree $\rho = 2$.

Construction 1: Assume that the storage size α and the reconstruction degree K are given. Determine the values of n and θ such that $n > K$ and $\rho\theta = n\alpha$. We will use the values $\beta_0 = \lceil \frac{\alpha}{2} \rceil$ and $\beta_1 = \alpha - \beta_0$ in this construction. Let $M = \alpha + \sum_{i=0}^{K-2}(\alpha - \beta_{i \pmod 2})$. Choose a (θ, M) MDS code. Now, generate a cycle graph with (initially empty) n vertices $V_0, V_1, ..., V_{n-1}$. In the following, we use the index $i = 0, 1, \ldots, n-1 \pmod n$. We denote n edges of the cycle graph by $e_0, e_1, \ldots, e_{n-1}$, and we use the convention that the vertex V_i has two incident edges e_{i-1} and e_i. Initially, we assign $\beta_{i \pmod 2}$ distinct symbols from $\{0,1,...,\theta-1\}$ into the edge e_i for all i so that no symbol is assigned to two different edges. Finally, each vertex V_i will contain $\beta_0 + \beta_1 = \alpha$ symbols which have been assigned to its two incident edges e_i and e_{i-1}. Then the FR code is the collection of $V_0, V_1, \ldots, V_{n-1}$, all of which are α-subsets of $\{0,1,...,\theta-1\}$.

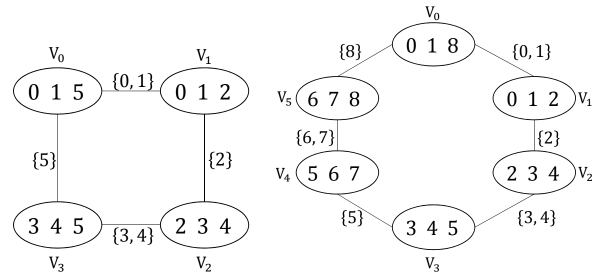


Fig. 5 Graph representations of FR codes with locality 2. (a) A $((6,6),3,(4,3,2))$ FR code with locality 2, (b) A $((9,6),3,(6,3,2))$ FR code with locality 2.

Example 1: Let's construct an FR code with locality 2 and $\rho = 2$ when $\alpha = 3$ and $K = 3$ is given. In this case, a file of size $M = 6$ can be stored. Determine the number n of storage nodes such that $n > K$ and n is even. If we select the value of n as 4, then θ should be 6. Since $\theta = M$, no MDS encoding is included in this case. And just assign these 6 symbols according to the above construction. Then we can obtain a $((6,6),3,(4,3,2))$ FR code with locality 2 as shown in Fig. 5(a).

If we select larger n, then we can improve the reliability of the system. If $n = 6$, then the number θ of coded symbols of MDS code should be 9. Then generate 9 symbols by a $(9,6)$ MDS code. And following the Construction 1, a $((9,6),3,(6,3,2))$ FR code with locality 2 in Fig. 5(b) can be obtained.

In both codes, every node can be repaired by its two adjacent nodes. For example, in a code of Fig. 5(a), a node storing V_0 can be repaired by downloading $\{0,1\}$ and $\{5\}$ from V_1 and V_3, respectively. In both codes, a data collector contacting any 3 storage nodes can retrieve 6 distinct symbols. They could reconstruct the data file of size 6.

Example 2: Let's construct an FR code with locality 2 and $\rho = 2$ when $\alpha = 4$ and $K = 3$ is given. In this case, a file of size $M = 8$ can be stored. Determine the number n of the storage nodes such that $n > 3$.

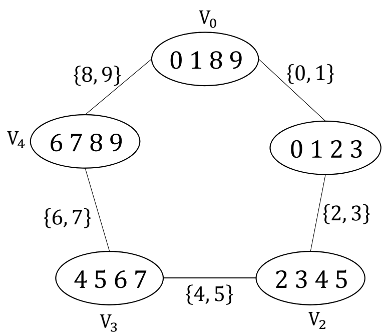


Fig. 6 A graph representation of $\big((10,8),3,(5,4,2)\big)$ FR code with locality 2

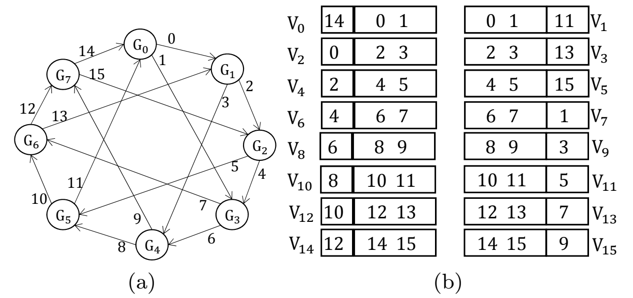


Fig. 7 An optimal $((16,10),6,(16,3,3))$ FR code with locality 2. (a) A 4-regular graph with 8 vertices and girth 4. (b) The distribution of 16 symbols by Construction 2.

Figure 6 is an example when $n=5$. Here, n could be an odd value unlike the example 1 since n can be any value which satisfies $2\theta = 4n$. Then θ should be 10. Figure 6 is a $\big((10,8),3,(5,4,2)\big)$ FR code with locality 2 that stores a file of size $M=8$ in 5 storage nodes of size 4. In this case, α is even, therefore, $\beta_0=\beta_1=\frac{\alpha}{2}$.

Theorem 2: A code $\mathcal{C}$ generated from Construction 1 is an optimal $((\theta,M),K,(n,\alpha,2))$ FR code with locality 2, where $2\theta=n\alpha$.

proof:

Construction 1 generates a code based on a cycle graph of length $n>K$. The upper bound on the capacity of the FR codes with locality 2 is obtained as the maximum file size of this code in Theorem 1. Therefore, it is obvious that the code $\mathcal{C}$ is an optimal $((\theta,M),K,(n,\alpha,2))$ FR code with locality 2. □

Remark 2: If we choose the number n of the storage nodes as $n=K+1$, then no MDS code is included in the FR code. If $n>K+1$ is selected, an MDS code can improve the reliability of the system.

Remark 3: When some two nodes fail simultaneously, they cannot be repaired locally. In the example 1, when two nodes storing V_0 and V_1 fail, local repair is not possible in both codes in Fig. 5(a) and 5(b). A $((9,6),3,(6,3,2))$ FR code can repair the failed nodes by downloading 6 symbols from any $K=3$ surviving storage nodes. The file of size $M=6$ can be obtained by decoding of the $(9,6)$ MDS code. Then symbols that will be contained in V_0 and V_1 should be generated by the encoding of the MDS code. In this case, the repair is possible but it is very complex unlike the local repair.

To guarantee the local-repair for all the two-node failures, the repetition degree ρ should be increased.

3.3 Construction for FR codes with locality 2 when $\rho=3$

In this subsection, we will discuss the FR codes with locality 2 when $\rho=3$. We will give a construction for $\alpha=3$ only. Therefore, the value of β_0 and β_1 is assumed to be 2 and 1, respectively, from (6).

Following construction is entirely dependent on the existence of a 4-regular graph with given girth. The existence of a d-regular graph with girth g is not immediately obvious, and it was first proved in [22] by some constructions. Since the number of storage nodes is related to the number of vertices of such a graph, it is important to find the graph with small number of vertices. This will be discussed at the next subsection.

Construction 2: Assume that the storage size $\alpha=3$ and the reconstruction degree K are given. Generate a 4-regular graph $\mathcal{G}$ with v vertices $G_0,G_1,...,G_{v-1}$, and girth $g\geq\lceil\frac{K+2}{2}\rceil$. We will use the values $\beta_0=2$ and $\beta_1=1$ in this construction. Let $M=3+\sum_{i=0}^{K-2}(3-\beta_{i \pmod 2})$. Choose a $(2v,M)$ MDS code. Each vertex of $\mathcal{G}$ has 4 edges, and we assign directions to them so that two edges are outgoing and the remaining two edges are incoming. Note that the number of edges is $2v$. Assign the symbols from $\{0,1,...,2v-1\}$ to $2v$ edges in one-to-one manner. Now we will distribute the symbols from $\{0,1,...,2v-1\}$ to initially empty sets $V_0,V_1,...,V_{2v-1}$ as follows:

1. For each $i=0,1,...,v-1$, two symbols assigned to two outgoing edges of G_i come into both V_{2i} and V_{2i+1} so that $|V_{2i}\cap V_{2i+1}|=2$.
2. For each $i=0,1,...,v-1$, two symbols assigned to two incoming edges of G_i are split and distributed to V_{2i} and V_{2i+1} individually so that $|V_{2i}|=|V_{2i+1}|=3$.

Then the collection of 3-subsets $V_0,V_1,...,V_{2v-1}$ is the FR code with parameters $((2v,M),K,(2v,3,3))$.

Example 3: Figure 7 shows an optimal $((16,10),6,(16,3,3))$ FR code with locality 2. This code is based on 4-regular graph with 8 vertices and girth 4.

Theorem 3: Construction 2 generates an optimal $((\theta,M),K,(n,3,3))$ FR code $\mathcal{C}$ with locality 2, where $\theta=n$. Moreover every storage node is guaranteed to have 2 disjoint repair sets.

proof: Every vertex G_i of a graph $\mathcal{G}$ has two incoming

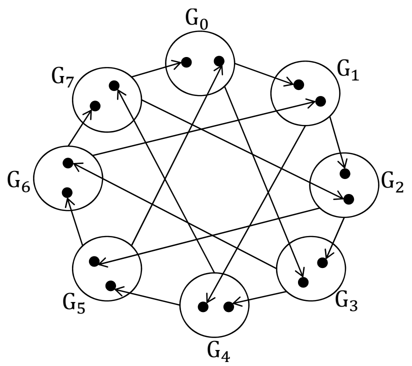


Fig. 8 A detailed graph $\mathcal{G}'$ of the graph $\mathcal{G}$ in Fig. 7.

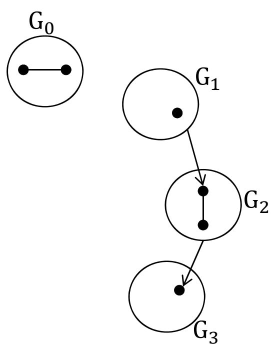


Fig. 9 A subgraph of $\mathcal{G}'$ induced by 6 storage nodes.

edges and two outgoing edges. Assume that (j_1, i) and (j_2, i) are two incoming edges and (i, k_1) and (i, k_2) are two outgoing edges of G_i.

Consider the storage node V_{2i} that is included in a vertex G_i. The first symbol of a node V_{2i} is stored in both V_{2j_1} and V_{2j_1+1}. The second and third symbols of V_{2i} are stored in the node V_{2i+1} in the same vertex G_i. One of these two symbols is stored in V_{2k_1} (or V_{2k_1+1}). And the other symbol is stored in V_{2k_2} (or in V_{2k_2+1}).

Then, every storage node V_{2i} has two disjoint repair sets, $\{V_{2j_1}, V_{2i+1}\}$ and $\{V_{2j_1+1}, V_{2k_1}, V_{2k_2}\}$. Similarly, every node V_{2i+1} also has two disjoint repair sets. Since the cardinality of the smallest repair set is 2, the code $\mathcal{C}$ has locality 2.

Now, we will prove the optimality of the code $\mathcal{C}$. Figure 8 shows the detailed graph $\mathcal{G}'$ of the 4-regular graph in Fig. 7(a). In $\mathcal{G}'$, every vertex G_i has two storage nodes, V_{2i} and V_{2i+1}. Every storage node is represented by a black dot in the graph. The first dot and the second dot in G_0 is the storage node V_0 and V_1, respectively. The edge that starts from a vertex G_0 and ends at V_2 in a vertex G_1 represents the symbol 0 which is the intersection among three storage nodes V_0, V_1, and V_2. Two storage nodes V_{2i} and V_{2i+1} in the same vertex G_i have two common symbols. We omit the edges between these two nodes V_{2i} and V_{2i+1} in the graph $\mathcal{G}'$ in Fig. 8.

If we select K storage nodes, then we can obtain a subgraph induced by the K nodes. Figure 9 shows a subgraph of $\mathcal{G}'$ induced by 6 storage nodes, V_0, V_1, V_3, V_4, V_5, and V_6. In a graph $\mathcal{G}'$ in Fig. 8, an edge starts from a vertex G_i and ends at a storage node V_j. This means that the edge is starts from both storage nodes V_{2i} and V_{2i+1} in a vertex G_i. Therefore, at least one of these two storage nodes and the storage node V_j is selected in a K-subset S, then an edge starts from G_i and ends at V_j is also included in the subgraph induced by the K storage nodes. In Fig. 9, the K-subset $S = \{0, 1, 3, 4, 5, 6\}$ is assumed. The edge starts from G_1 and ends at V_4 is included in the subgraph since one storage node V_3 of G_1 and the storage node V_4 is contained in S. An edge starts from G_2 and ends at V_6 is also included in the subgraph. Edges between V_0, V_1 and V_3, V_4 are also implied. These edges represents two symbols that are the intersection of two storage nodes.

Let's denote the subgraph of $\mathcal{G}'$ induced by S as $\mathcal{H}_S$. In the subgraph $\mathcal{H}_S$, edges with direction represents the one symbol intersection between G_i and G_j and the other edge represents the two symbol intersection between two storage nodes V_{2i} and V_{2i+1}. Let $E' = E'_1 \cup E'_2$ be a set of edges in $\mathcal{H}_S$, where E'_1 is a subset of edges with direction and E'_2 is the set of the remaining edges in $\mathcal{H}_S$. Now we can calculate the cardinality of the union of K storage nodes in S as follows.

$$M(K) = K\alpha - \Big(\sum_{e \in E'_1} 1 + \sum_{e \in E'_2} 2\Big).$$

Now we should determine the K-subset S which gives the minimum $M(K)$. To minimize the $M(K)$, the cardinality $|E'_2|$ must be maximized first. Therefore, as many pairs of two storage nodes V_{2i} and V_{2i+1} which are in the same vertex G_i should be included in S as possible. $|E'_2| = \lfloor \frac{K}{2} \rfloor$ is the maximum possible $|E'_2|$ for given K. And then $|E'_1|$ must be maximized for given E'_2. Then $|E'_1| = \lceil \frac{K}{2} \rceil - 1$ is the maximum possible $|E'_1|$ for given K and $|E'_2| = \lfloor \frac{K}{2} \rfloor$. As a special case, $|E'_1| = \lceil \frac{K}{2} \rceil$ is possible if the graph $\mathcal{G}$ of a code $\mathcal{C}$ has a cycle of length $\lceil \frac{K}{2} \rceil$ or less.

In Construction 2, the code $\mathcal{C}$ is generated based on the graph with girth $g \geq \lceil \frac{K+2}{2} \rceil$. Therefore, in the code $\mathcal{C}$, $|E'_2| \leq \lfloor \frac{K}{2} \rfloor$ and $|E'_1| \leq \lceil \frac{K}{2} \rceil - 1$. Then the maximum file size is

$$\begin{aligned} M(K) &= K\alpha - \big(2|E'_2| + 1|E'_1|\big) \\ &\geq 3K - \Big(2\Big\lfloor \frac{K}{2} \Big\rfloor + \Big\lceil \frac{K}{2} \Big\rceil - 1\Big). \end{aligned}$$

The right-hand side of the above inequality is $\mathcal{M}(K, 3)$. Therefore, the code $\mathcal{C}$ is an optimal $((2v, \mathcal{M}(K, 3)), K, (2v, 3, 3))$ FR code with locality 2.

□

3.4 Some practical considerations

Construction 2 depends on the existence of 4-regular graphs with a given girth g. It is important to obtain such a graph with small number v of vertices since v determines the number n of the storage nodes as $n = 2v$.

Moore bound is a simple lower bound on the number of vertices for a d-regular graph with a given girth.

Proposition 1: (Moore bound) [20] The number $v(d,l)$ of vertices in a d-regular graph of girth g is lower bounded by $v_0(d,l)$, where

$$v_0(d,2m+1) = 1 + d \cdot \sum_{i=0}^{m-1} (d-1)^i, \qquad v_0(d,2m) = 2 \cdot \sum_{i=0}^{m-1} (d-1)^i. \tag{9}$$

The d-regular graphs with girth g and v_0 vertices in (9) are called *Moore graphs*. These graphs are unique if they exist [20]. Complete graphs K_{d+1} and complete bipartite graphs $K_{d,d}$ are Moore graphs with $g=3$ and $g=4$, respectively.

And for $g>4$, Moore graphs only exist if the pair of parameters (d,g) is $(3,5),(7,5)$, and $(57,5)$, or if $g=6,8$, or 12 [20].

The d-regular graphs of girth g which have the smallest possible number of vertices are called (d,g) *cages*. The number of vertices $v_c(d,g)$ of (d,g) cage is the same as $v_0(d,g)$ if Moore graphs with the parameter (d,g) exist. Otherwise, $v_c(d,g) > v_0(d,g)$.

If we use the $(4,g)$ cages for $g \geq \lceil \frac{K+2}{2} \rceil$ in Construction 2, then we can obtain an optimal $((\theta,\mathcal{M}),K,(n,3,3))$ FR code with locality 2 in terms of the storage overhead which is the fraction of total storage size $n\alpha$ and the file size M. However, very few cages are known [19].

We can find some explicit 4-regular graphs with girth g, where $g \leq 6$ in [21]. The smallest number of vertices of a 4-regular graph with girth 4 is 8. If we use this graph in Construction 2, the required number of storage nodes is 16 and this guarantees that the resulting code achieves the capacity for $K \leq 6$.

Table 1 shows the known 4-regular graphs with girth g, for $3 \leq g \leq 10$, and $g=12$, and parameters of the FR code obtained by the Construction 2 using the graph. In this table, v is the number of vertices required for a 4-regular graph with girth g. n denotes the number of storage nodes of a code which is constructed from the graph with v vertices and it is obvious that $n=2v$ from Construction 2. The graph with girth $3,4,6,8$, and 12 are the Moore graphs while graph with girth $5,7,9$ and 10 are not.

If we want to store a file of large size, then we select either large α or large K since $\mathcal{M}(K,\alpha)$ is increasing with K and α. Construction 2 is only for a code with fixed $\alpha=3$. Therefore, we must choose large K to store a file of large size. The value of K is also restricted by the girth g of a 4-regular graph that is used in the construction. The generation of 4-regular graphs with large girth and small number of vertices is not straightforward.

Table 1 Known $(4,g)$ cages with girth g [23]

g	v	Moore bound	n	K_{max}	$\mathcal{M}(K_{max})$	storage overhead
3	5	√	10	4	7	30/7
4	8	√	16	6	10	48/10
5	19		38	8	13	114/13
6	26	√	52	10	16	156/16
7	67		134	12	19	402/19
8	80	√	160	14	22	480/22
9	275		550	16	25	1650/25
10	384		768	18	28	2304/28
12	728	√	1456	22	34	4368/34

Both an FR code with $\alpha=11$, $K=3$ and an FR code with $\alpha=3$, $K=14$ can store a file of size 22. Since Construction 2 only gives a code with $\alpha=3$, we should construct an FR code with $\alpha=3$, $K=14$ to store a file of size 22. Such an FR code can be constructed based on a 4-regular graph with girth $g=8$. If the file size is increased much larger, then a 4-regular graph with much larger girth is required. Only $(4,g)$ cages with $3 \leq g \leq 10$, and $g=12$ are known. Cages with other girth are not known. Furthermore, although a Moore graph exists for large girth g and is used in Construction 2, the number of required vertices increases much larger when the girth increases by 1.

If we want to store a file of size 28, there are several options. One is to use a large code with $K=18$ in the 8-th row of Table 1. This requires 768 storage nodes. The other is to use four codes with $K=4$ in the 1-st row of Table 1. This requires 40 storage nodes. The latter seems to be much efficient. Therefore, using multiple disjoint small codes could be another good option for storing large file.

4. Concluding Remarks

In this paper, we proposed explicit constructions for FR codes with locality 2 when $\rho=2$ or $\rho=3$. The construction for $\rho=2$ is applicable for general values of α while the construction for $\rho=3$ is restricted to the case of $\alpha=3$ only. Constructions for general values of $\alpha>3$ when the repetition degree $\rho=3$ will be an interesting future work.

The proposed constructions are based on regular graphs, especially 4-regular graphs for $\rho=3$. To construct a code which can store a large file, $(4,g)$ cages with large girth g will be the best choice. Such a $(4,g)$ cages are known for only small ranges of g. Construction for $(4,g)$ cages for all positive integers g not shown in Table 1 will be an important and interesting research.

References

[1] A.G. Dimakis, P.B. Godfrey, M. Wainwright, and K. Ram-

chandran, "Network coding for distributed storage systems," in *Proc. 26th IEEE Int. Conf. Comput. Commun.*, AK, May 2007, pp.2000-2008.

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

[3] P. Gopalan, C. Huang, H. Simitci, and S. Yekhanin, "On the locality of codeword symbols," *IEEE Trans. Inf. Theory*, vol. 58, no. 11, pp. 6925-6934, Nov. 2012.

[4] A.S. Rawat, D.S. Papailiopoulos, and A.G. Dimakis, "Availability and locality in distributed storage," in *Proc. Int. Symp. Inf. Theory (ISIT)*, Honolulu, Jun./Jul. 2014, pp. 681-685.

[5] I. Tamo and A. Barg, "Bounds on locally recoverable codes with multiple recovering sets," in *Proc. Int. Symp. Inf. Theory (ISIT)*, Honolulu, Jun./Jul. 2014, pp. 691-695.

[6] K. V. Rashmi, N. B. Shah, P. V. Kumar, and K. Ramchandran, "Explicit construction of optimal exact regenerating codes for distributed storage," in *Proc. 47th Annu. Allerton Conf. Commun., Control, Comput. (Allerton)*, Sept./Oct. 2009, pp. 1243-1249.

[7] N. B. Shah, K. V. Rashmi, P. V. Kumar, and K. Ramchandran, "Distributed storage codes with repair-by-transfer and nonachievability of interior points on the storage-bandwidth tradeoff," *IEEE trans. Inform. Theory* vol. 58, no. 3, pp. 1837-1852.

[8] 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. Inform. Theory*, vol. 58, no. 4, pp. 2134-2158, Apr. 2012.

[9] S.E. Rouayheb and K. Ramchandran, "Fractional repetition codes for repair in distributed storage systems," presented at the *48th Annu. Allerton Conf. Control, Comput, Commun.*, Urbana-Champaign, IL, Sep. 2010.

[10] S. Pawar, N. Noorshams, S. E. Rouayheb, and K. Ramchandran, "DRESS codes for the storage cloud: simple randomized construction," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jul./Aug. 2011, pp. 2338-2342.

[11] O. Olmez and A. Ramamoorthy, "Repairable replication-based storage systems using resolvable designs," in *Proc. 50th Annu. Allerton Conf. Commun., Control, Comput. (Allerton)*, Oct. 2012, pp. 1174-1181.

[12] O. Olmez and A. Ramamoorthy, "Replication based storage systems with local repair," in *Proc. Int. Symp. Netw. Coding (NetCod)*, 2013, pp. 1-6.

[13] J. C. Koo and J. T. Gill, "Scalable constructions of fractional repetition codes in distributed storage systems," in *Proc. 49th Annu. Allerton Conf. Commun., Control, Comput. (Allerton)*, Sept. 2011, pp. 1366-1373.

[14] B. Zhu, K. W. Shum, H. Li, and H. Hou, "General fractional repetition codes for distributed storage systems," *IEEE Commun. Lett.*, vol. 18, no. 4, pp. 660-663, Apr. 2014.

[15] B. Zhu, and H, Li, "Adaptive fractional repetition codes for dynamic storage systems," *IEEE Comm. Lett.*, vol. 19, no. 12, pp. 2078-2081, Dec. 2015.

[16] N. Silberstein, and T. Etzion, "Optimal fractional repetition codes based on graphs and designs," *IEEE Trans. Inf. Theory*, vol. 61, no. 8, pp. 4164-4180, Aug. 2015.

[17] M.-Y. Nam, J.-H. Kim, and H.-Y. Song, "Locally repairable fractional repetition codes," in *Proc. 7th Int. Workshop on Signal Design and its Applications in Commun. (IWSDA)*, Sept. 2015, pp. 128-132.

[18] A. Barg and G. Zemor, "Concatenated codes: serial and parallel," in *IEEE Trans. Inf. Theory*, vol. 51, no. 5, pp. 1625-1634, May 2005.

[19] N. Biggs, "Algebraic graph theory," *Cambridge Univ. Press*, 1993.

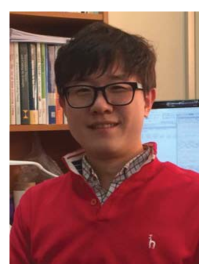
[20] D. Jakobson and I. Rivin, "On some extremal problems in graph theory," *arXiv:math.CO/9907050.*

[21] M. Meringer, "Fast generation of regular graphs and construction of cages," *Journal of Graph Theory*, vol. 30, no. 2, pp. 137-146, Feb. 1999.

[22] H. Sachs, "Regular graphs with given girth and restricted circuits," *J. London Math. Soc.*, 38, pp. 423-429, 1963.

[23] G. Royle, "Cages of Higher Valency," http://school.maths.uwa.edu.au/~gordon/remote/cages/allcages.html.

Mi-Young Nam received the BS and the MS degree in Electrical and Electronics Engineering from Yonsei University of Seoul, Korea, in 2005 and 2009, respectively. She was with the Samsung Electronics Co., Ltd., Suwon, Korea as an engineer from 2005 to 2007. She is currently working toward her Ph.D degree in Electrical and Electronics Engineering at Yonsei University. Her area of research interest includes error-correcting codes, codes for distributed storage systems, especially locally repairable codes.



Jung-Hyun Kim received the BS and the MS degree in Electrical and Electronics Engineering from Yonsei University of Seoul, Korea, in 2006 and 2008, respectively. He spent 3 years as a member of Electronics and Telecommunications Research Institute (ETRI), Daejeon, Korea, from 2010 to 2012, contributed to a team developing a fundamental technology on next-generation wireless mesh communication system. He is currently working toward his Ph.D degree in Electrical and Electronics Engineering at Yonsei University. His area of research interest includes distributed storage systems, coding theory, and applications of wireless communications.

Hong-Yeop Song received his B.S. degree in Electronic Engineering from Yonsei University in 1984, MSEE and Ph.D. degrees from the University of Southern California, Los Angeles, California, in 1986 and 1991, respectively. He spent 2 years as a research associate at USC and then 2 years as a senior engineer at the standard team of Qualcomm Inc., San Diego, California. Since Sept. 1995, he has been with Dept. of electrical and electronic engineering, Yonsei University, Seoul, Korea. His area of research interest includes digital communications and channel coding, design and analysis of various pseudo-random sequences for communications and cryptography. He is a member of IEICE, IEEE, MAA, KICS, KIISC and KMS.