Fractional repetition codes with flexible repair from combinatorial designs

Oktay Olmez and Aditya Ramamoorthy

Abstract

Fractional repetition (FR) codes are a class of regenerating codes for distributed storage systems with an exact (table-based) repair process that is also uncoded, i.e., upon failure, a node is regenerated by simply downloading packets from the surviving nodes. In our work, we present constructions of FR codes based on Steiner systems and resolvable combinatorial designs such as affine geometries, Hadamard designs and mutually orthogonal Latin squares. The failure resilience of our codes can be varied in a simple manner. We construct codes with normalized repair bandwidth ($\beta$) strictly larger than one; these cannot be obtained trivially from codes with $\beta = 1$. Furthermore, we present the Kronecker product technique for generating new codes from existing ones and elaborate on their properties. We also present local FR codes where the repair degree is smaller than the number of nodes contacted for reconstructing the stored file. For these codes a tradeoff between the local repair property and failure resilience is established and codes that meet this tradeoff are constructed. Much of prior work only provided lower bounds on the FR code rate. In our work, for most of our constructions we determine the code rate for certain parameter ranges.

Index Terms

classification: fractional repetition codes, combinatorial designs, Steiner systems, affine geometry, high girth, resolvable designs, regenerating codes, local repair.

I. INTRODUCTION

Large scale data storage systems that are employed in social networks, video streaming websites and cloud storage are becoming increasingly popular. In these systems, the integrity of the stored data and the speed of the data access needs to be maintained even in the presence of unreliable storage nodes. This issue is typically handled by introducing redundancy in the storage system, through the usage of replication and/or erasure coding. However, the large scale, distributed nature of the systems under consideration introduces another issue. Namely, if a given storage node fails, it needs to be regenerated so that the new system continues to have the properties of the original system. It is of course desirable to perform this regeneration in a distributed manner and optimize performance metrics associated with the regeneration process. Firstly, one would like to ensure that the regeneration process be fast. For this purpose we would like to minimize the data that needs to be downloaded from the surviving nodes. Moreover, we would like the surviving nodes and the new node to perform very little (ideally no) computation, as
this also induces a substantial delay in the regeneration process that is comparable to the download time (see [1]).

In addition, the regeneration induces a workload on the surviving storage nodes and it is desirable to perform the regeneration by connecting to a small number of nodes. Connecting to a small set of nodes also reduces the overall energy consumption of the system.

In recent years, codes which are designed to satisfy the needs of data storage systems have been the subject of much investigation and there is extensive literature on the topic. Depending upon the specific metrics that are optimized there are different requirements on the distributed storage system. However, broadly speaking all systems have the following general characteristics. A distributed storage system (henceforth abbreviated to DSS) consists of \(n\) storage nodes, each of which stores \(\alpha\) packets (we use symbols and packets interchangeably). A given user, also referred to as the data collector needs to have the ability to reconstruct the stored file by contacting any \(k\) nodes; this is referred to as the maximum distance separability (MDS) property of the system. To ensure the reliability in the system, the DSS also needs to repair a failed node. This is accomplished by contacting a set of \(d\) surviving nodes and downloading \(\beta\) packets from each of them for a total repair bandwidth of \(\gamma = d\beta\) packets. Thus, the system has a repair degree of \(d\), normalized repair bandwidth \(\beta\) and total repair bandwidth \(\gamma\). The new DSS should continue to have the MDS property.

A simple technique for obtaining a distributed storage system is to treat the file that needs to be stored as a set of symbols over a large alphabet, generate encoded symbols by using an MDS code (such as a Reed-Solomon (RS) code) and then store each encoded symbol on a different storage node. It is well recognized that the drawback of this method is that upon failure of a given storage node, a large amount of data needs to be downloaded from the remaining storage nodes (equivalent to recreating the file). To address this issue, the technique of regenerating codes was developed in the work of Dimakis et al. [2]. In the framework of [2], the repair degree \(d \geq k\) and the system needs to have the property that a failed node can be repaired from any set of \(d\) surviving nodes. The principal idea of regenerating codes is to use subpacketization. In particular, one treats a given physical block as consisting of multiple symbols (unlike the MDS code that stores exactly one symbol in each node). Coding is now performed across the packets such that the file can be recovered by contacting a certain minimum number of nodes. In addition, one can regenerate a failed node by downloading appropriately coded data from the surviving nodes. The work of [2] identified a fundamental tradeoff between the amount of storage at each node and the amount of data downloaded for repairing a failed node under the mechanism of functional repair, where the new node is functionally equivalent to the failed node, though it may not be exact copy of it. Two points on the curve deserve special mention and are arguably of the most interest from a practical perspective. The minimum bandwidth regenerating (MBR) point refers to the point where the repair bandwidth, \(\gamma\) is minimum. Likewise, the minimum storage regenerating (MSR) point refers to the point where the storage per node, \(\alpha\) is minimum.

In a different line of work, it has been argued that repair bandwidth is not the only metric for evaluating the repair process. It has been observed that the number of nodes that are contacted for purposes of repair is also an important metric that needs to be considered. The model of [2], which enforces repair from any set of \(d\) surviving nodes requires \(d\) to be at least \(k\). The notion of local repair was introduced in [3], [4], [5], and considers the design of DSS where \(d < k\). However, one only requires that there is some set of \(d\) surviving nodes from which the repair can take place.

The majority of work in the design of codes for DSS considers coded repair where the surviving nodes and the new node need to compute linear combinations for regeneration. It is well recognized that the read/write bandwidth
The process of contacting the surviving nodes induces additional undesirable delays in the repair process. The process can also be potentially memory intensive since the packets comprising the file are often very large (of the order of GB). Motivated by these issues, reference [7] considered the following variant of the DSS problem. The DSS needs to satisfy the property of exact and uncoded repair, i.e., the regenerating node needs to produce an exact copy of the failed node by simply downloading packets from the surviving nodes. This allows the entire system to work without requiring any computation at the surviving nodes. In addition, they considered systems that are resilient to multiple (>1) failures. However, the DSS only has the property that the repair can be conducted by contacting some set of d nodes, i.e., unlike the original setup, repair is not guaranteed by contacting any set of d nodes. This is reasonable as most practical systems operate via a table-based repair, where the new node is provided information on the set of surviving nodes that it needs to contact. The work of [7] proposed a construction whereby an outer MDS code is concatenated with an inner “fractional repetition” code that specifies the placement of the coded symbols on the storage nodes. The main challenge here is to design the inner fractional repetition code in a systematic manner.

In this work, we present several families of FR codes and analyze their properties.

A. Main Contributions

- We construct a large class of FR codes based on resolvable designs [8] where the repetition degree (ρ) of the symbols can be varied in an easy manner. The constructions of [7] lack this flexibility as they are mostly based on Steiner systems where the repetition degree is usually fixed by the construction.
- We consider FR codes where β > 1, i.e., the new node downloads more than one packet from the d surviving nodes. We emphasize that starting with a FR code with β = 1, it is trivially possible to arrive at a code with β > 1 simply by replacing each existing symbol with multiple new symbols. However, such a strategy only results in a limited range of system parameters that can be achieved. We present several examples to show that our proposed codes achieve certain parameter ranges that cannot be achieved in a trivial manner.
- Determining the file size that can be supported by a given FR code turns out be challenging. Much of the literature in combinatorial designs only discusses the pairwise overlaps between the content of the different storage nodes. However, the file size depends on the union of all subsets of storage nodes of size k. In this work we determine the file sizes for most of our constructions. In particular, we demonstrate a family of FR codes whose file size is strictly larger than a simple lower bound that is obtained by applying the inclusion-exclusion principle. We also determine the file size for a large class of codes obtained from Steiner systems that were originally considered in [7]. Several of our constructions are shown to meet the Singleton bound for specific file sizes, which demonstrates their optimality.
- We present the Kronecker product as a technique for constructing new FR codes from existing ones and analyze the properties of codes thus obtained.
- In this work, we propose a large family of locally recoverable FR codes where d < k. These codes enjoy the benefit of repair by connecting to a small number of nodes. We derive an appropriate minimum distance bound for our class of codes that enjoy local, exact and uncoded repair, and demonstrate constructions that meet these bounds.

This paper is organized as follows. In Section II, we outline our precise problem formulation and elaborate on the related work in the literature. We discuss our FR code constructions for the case when d ≥ k in Section III.
Fig. 1. A DSS with $(n, k, d, \alpha) = (5, 3, 4, 4)$. Each node contains a subset of size 4 of the packets from \{b_1, \ldots, b_{10}\}. Node $V_1$ for instance contains symbols $b_i$, $i = 1, \ldots, 4$ that is $V_1 = \{b_1, b_2, b_3, b_4\}$. When there is no confusion we simply use the notation $V_1 = b_1b_2b_3b_4$ instead of the set notation $V_1 = \{b_1, b_2, b_3, b_4\}$.

and explain the Kronecker product technique in Section IV. The locally recoverable FR codes where $d < k$ are considered in Section V and Section VI outlines the conclusions and future work.

II. BACKGROUND AND RELATED WORK

A DSS is specified by parameters $(n, k, d, \alpha)$ where $n$ - number of storage nodes, $k$ - the minimum number of nodes to be contacted for recovering the file and $d$ - the number of nodes to be contacted in order to regenerate a failed node and $\alpha$ - the storage capacity. In case of repair, the new node downloads $\beta$ packets from each surviving node, for a total of $\gamma = d\beta$ bits. Let $M$ denote the size of file being stored on the DSS. We consider the design of fractional repetition codes that are best explained by means of the following example [9] with $(n, k, d, \alpha) = (5, 3, 4, 4)$ in the discussion below.

Example 1: Consider a file of $M = 9$ packets $(a_1, \ldots, a_9) \in \mathbb{F}_q^9$ that needs to stored on the DSS. We use a (10, 9) MDS code that outputs 10 packets $b_i = a_i$, $i = 1, \ldots, 9$ and $b_{10} = \sum_{i=1}^{9} a_i$. The coded packets $b_1, \ldots, b_{10}$ are placed on $n = 5$ storage nodes as shown in Fig. 1. This placement specifies the inner fractional repetition code. It can be observed that each $b_i$ is repeated $\rho = 2$ times and the total number of symbols $\theta$ = 10. Any user who contacts any $k = 3$ nodes can recover the file (using the MDS property). Moreover, it can be verified that if a node fails, one packet each can be downloaded from the four surviving nodes, i.e., $\beta = 1$ and $d = 4$, so that $\gamma = 4$.

Thus, the approach uses an MDS code to encode a file consisting of a certain number of symbols. Let $\theta$ denote the number of encoded symbols. Copies of these symbols are placed on the $n$ nodes such that each symbol is repeated $\rho$ times and each node contains $\alpha$ symbols. Moreover, if a given node fails, it can be exactly recovered by downloading $\beta$ packets from some set of $d$ surviving nodes, for a total repair bandwidth of $\gamma = d\beta$. It is to be noted that in this case $\alpha = \gamma$, i.e., these schemes operate at the MBR point. In the example above, $\beta = 1$, so that $\alpha = d$. One can also consider systems with $\beta > 1$ in general. A simple way to do this is replicating the symbols in the storage system. The resultant DSS has the parameters $(n, k, d, \beta\alpha)$ with $\beta > 1$. However there are infinite families of FR codes with $\beta > 1$ which cannot be obtained this way. In Fig. 2 we illustrate the DSS obtained by replicating the $(5, 3, 4, 4)$-DSS when $\beta = 2$.

Before introducing the formal definition of a fractional repetition (FR) code we need the notion of $\beta$-recoverability. Let $[n]$ denote the set $\{1, 2, \ldots, n\}$.
Definition 1: Let $\Omega = [\theta]$ and $V_i, i = 1, \ldots, d$ be subsets of $\Omega$. Let $V = \{V_1, \ldots, V_d\}$ and consider $A \subseteq \Omega$ with $|A| = d\beta$. We say that $A$ is $\beta$-recoverable from $V$ if there exists $B_i \subseteq V_i$ for each $i = 1, \ldots, d$ such that $B_i \subseteq A$, $|B_i| = \beta$ and $\bigcup_{i=1}^{d} B_i = A$.

Definition 2: A fractional repetition (FR) code $C = (\Omega, V)$ for a $(n, k, d, \alpha)$-DSS with repetition degree $\rho$ and normalized repair bandwidth $\beta = \alpha/d$ ($\alpha$ and $\beta$ are positive integers) is a set of $n$ subsets $V = \{V_1, \ldots, V_n\}$ of a symbol set $\Omega = [\theta]$ with the following properties.

(a) The cardinality of each $V_i$ is $\alpha$.
(b) Each element of $\Omega$ belongs to $\rho$ sets in $V$.
(c) Let $V_{\text{surv}}$ denote any $(n - \delta)$ sized subset of $V$ and $V_{\text{fail}} = V \setminus V_{\text{surv}}$. Each $V_j \in V_{\text{fail}}$ is $\beta$-recoverable from some $d$-sized subset of $V_{\text{surv}}$. Let $\rho_{\text{res}}$ be the maximum value of $\delta$ such that this property holds.

The value of $\rho_{\text{res}}$ is a measure of the resilience of the system to node failures, under the constraint of exact and uncoded repair. The file size is given by

$$M = \min_{I \subseteq [n], |I| = k} |\bigcup_{i \in I} V_i|$$

and the code rate is defined as $R_C = \frac{M}{n\alpha}$. We emphasize that $R_C$ depends on $k$.

Note that the parameters of a FR code are such that $\theta\rho = n\alpha$. Thus, the code rate $R_C = \frac{M}{n\alpha} \leq \frac{\theta}{n\alpha} = \frac{1}{\rho}$. Moreover as $\rho \geq 2$, the maximum rate of any FR code is at most $\frac{1}{2}$. It is to be noted that the parameter $M$ also sets the code rate of the outer MDS code; it is exactly $M/\theta$. For a FR code $C = (\Omega, V)$ and an index set $I \subseteq [n]$, we say that nodes $V_i \in V$ for $i \in I$ cover $\zeta$ symbols if $\zeta = |\bigcup_{i \in I} V_i|$.

The work of [7], only considered FR codes with $\beta = 1$ and $k \leq d$, i.e., for recovery the new node would contact $d$ surviving nodes and download a single packet from each of them. In this case the requirement (c) in Definition 2 is automatically satisfied and it can be seen that the system is resilient to $\rho - 1$ failures, i.e., $\rho_{\text{res}} = \rho - 1$. It is to be noted that the requirement of $d \geq k$ is essential in the problem formulation considered in [2] since the systems require node recovery from any set of $d$ surviving nodes. In that setup if $d < k$, it is easy to see that one can always specify a failed node and a set of $d$ nodes from which recovery is impossible. However, in the framework of [7], the recovery requirement is relaxed. Specifically, to recover from a failure, the new node contacts a specific...
set of nodes from which it regenerates the failed node. Thus, the recovery process is table-based and for each node we only need to guarantee the existence of one set of \(d\) nodes from which recovery is possible. Thus, it becomes possible to have systems with \(d < k\). In fact, in Section \(\Box\) of this paper, we present several constructions of FR codes where \(d < k\). In the literature these are referred to as codes that allow for local repair.

For FR codes, the failure resilience \(\rho_{res}\) and the code rate \(R_C\) are two evaluation metrics and it is evident that there is a tradeoff between them. Indeed, if the outer MDS code does not add any redundancy, i.e., \(M = \theta\) then \(k\) would need to be chosen such that any \(k\) nodes cover all the \(\theta\) symbols and the code rate of the system would be exactly \(\frac{\theta}{n\alpha}\). However, in this case the DSS will be resilient to at most \(\rho - 1\) failures under any possible recovery procedure, i.e., even without any constraint on the repair. In contrast if the outer code introduces nontrivial redundancy, the file size \(M\) would be lower but it may be possible to reconstruct the DSS in the presence of more than \(\rho - 1\) failures. To see this, consider Example \(\square\) where the outer MDS code has rate \(9/10\). Note that under exact and uncoded repair, this DSS is resilient to only one failure. However, the DSS can be reconstructed even in the presence of the failure of any two nodes, since any three surviving nodes cover at least nine symbols. Our proposed codes will also be evaluated in terms of their minimum distance which quantifies this tradeoff.

**Definition 3: Minimum Distance.** The minimum distance of a DSS denoted \(d_{\min}\) is defined to be the size of the smallest subset of storage nodes whose failure guarantees that the file is not recoverable from the surviving nodes. The Singleton bound on the minimum distance is well known.

**Lemma 1 (Singleton Bound):** Consider a DSS with parameters \((n, k, d, \alpha)\) with file size \(M\) and minimum distance \(d_{\min}\). Then,

\[
d_{\min} \leq n - \left\lceil \frac{M}{\alpha} \right\rceil + 1.
\]

It turns out that codes that have the local repair property, i.e., codes with \(d < k\) suffer a penalty on the maximum possible minimum distance. This tradeoff was captured in the case of scalar (i.e., \(\alpha = 1\)) codes by [3] and by [4] in the case of vector (i.e., \(\alpha > 1\)) codes.

**Lemma 2:** Consider a DSS with parameters \((n, k, d, \alpha)\) with file size \(M\) and minimum distance \(d_{\min}\). Then,

\[
d_{\min} \leq n - \left\lceil \frac{M}{\alpha} \right\rceil + \left\lceil \frac{M}{d\alpha} \right\rceil + 2.
\]

It can be observed that if \(d \geq k\), then the bound above reduces to the Singleton bound.

**Observation 1:** We note that a given DSS meets the Singleton bound if \(k = \left\lceil \frac{M}{\alpha} \right\rceil\). Similarly, a code meets the bound in Lemma \(2\) if \(k = \left\lceil \frac{M}{\alpha} \right\rceil + \left\lceil \frac{M}{d\alpha} \right\rceil - 1\).

It is to be noted that the bound in Lemma \(2\) holds for all possible local repair codes. In this work, we consider the added constraint that the repair takes place purely by download. Thus, for our constructions, the bound in Lemma \(2\) is in general loose. In Section \(\Box\) we derive a tighter upper bound on the minimum distance of codes where the repair process is local and operates purely by download.

Many of our constructions will result from combinatorial designs that we briefly introduce (a detailed description can be found in [8]). A design is a pair \((\Omega, V)\) where \(\Omega\) is a set of elements called “points” and \(V\) is a collection of non-empty subsets of \(\Omega\) called “blocks”. A prototypical example with several applications is the balanced incomplete block design (BIBD).

**Definition 4:** A \((\theta, \rho, \alpha, \lambda)\) balanced incomplete block design (BIBD) is a pair \((\Omega, V)\), where \(\Omega\) is a \(\theta\)-element set called points and \(V\) is a collection of \(\alpha\)-subsets of \(\Omega\) called blocks, such that \(|V| = n\); every element of \(V\) is contained in exactly \(\rho\) blocks and every 2-subset of \(\Omega\) is contained in exactly \(\lambda\) blocks.
Let $n$ denote the number of blocks. It can be shown that for a BIBD, the following relations hold.

\[ n\alpha = \theta \rho, \quad \text{and} \quad \rho(\alpha - 1) = \lambda(\theta - 1). \]  

It can be observed that a $(\theta, \rho, \alpha, \lambda)$-BIBD can be used as the FR code in a DSS as long as $\beta$-recoverability is guaranteed for an appropriate $\beta$ (there are several instances when $\beta = 1$). These $(\theta, \rho, \alpha, \lambda)$-BIBDs include finite projective planes and affine planes. A $(a^2 + a + 1, a+1, a+1, 1)$-BIBD is equivalent to a projective plane of order $a$. Projective planes have interesting geometric properties that can be used in determining the corresponding file size. For instance, any two blocks of a $(a^2 + a + 1, a+1, a+1, 1)$-BIBD share exactly one point and any two points are contained in exactly one block in a projective plane. The smallest example of a projective plane corresponding to $a = 2$ is known as the Fano plane and is depicted in Fig. 3. For more information on the projective planes and affine planes we refer the Chapter 2 of [8].

![Figure 3](image.png)

Fig. 3. $(7, 3, 3, 1)$-BIBD also known as Fano plane.

One can use the Fano plane to design the inner FR code, by interpreting the points as symbols and the blocks as storage nodes. Suppose we first apply a $(7, 6)$-MDS to the file. Then we can place the coded symbols on the storage nodes as depicted in Fig. 3. Note that these storage nodes are obtained from the blocks. Since any two nodes share exactly one symbol, contacting any three nodes will recover the file. An affine plane can be obtained by deleting one block and its all points from a projective plane. Hence, an affine plane of order $a$ is equivalent to a $(a^2, a, a, 1)$-BIBD. Here any two points are contained in exactly one block, however any two blocks do not have to share any points. More generally an FR code can be obtained from Steiner systems.

**Definition 5:** A $S(t, \alpha, \theta)$ Steiner system is a set $\Omega$ of $\theta$ elements and a collection of subsets of $\Omega$ of size $\alpha$ called blocks such that any $t$-subset of the symbol set $\Omega$ appears exactly one of the blocks.

Steiner systems are examples of $t$-designs. An FR code is a $t$-design if every $t$-subset of symbols is contained in exactly $\lambda$ nodes. The concept of $t$-designs can be viewed as a generalization of the concept of BIBD’s. Naturally a $S(2, \alpha, \theta)$ Steiner system is a $(\theta, \rho, \alpha, 1)$-BIBD where

\[ \rho = \frac{\theta - 1}{\alpha - 1}, \quad n = \frac{(\theta - 1)\theta}{(\alpha - 1)\alpha}, \quad d = \alpha \quad \text{and} \quad \beta = 1. \]

Thus, projective planes and affine planes are instances of Steiner systems. A given FR code can be put in one-to-one correspondence with an incidence matrix as explained below.
Moreover, it can be observed that any two nodes recover at least 6 bound
from a single node failure by downloading two packets each from two nodes in the same column; hence
using an outer
with discussion we will refer to this technique for obtaining codes with
obtaining codes with
\( \beta > 1 \) meets the Singleton bound. However, this code (with
\( \beta = 1 \)) cannot be arrived at simply by replication. To see this
we note that if this were true, the original DSS with
\( \beta = 1 \) must correspond to a storage capacity of 2 and have a

**Definition 6:** The incidence matrix of a FR code \( C = (\Omega, V) \) where \( \Omega = [\theta] \) and \( V = \{V_1, V_2, \ldots, V_n\} \) is the
\( \theta \times n \) \{0,1\}-matrix \( N \) defined by
\[
N_{i,j} = \begin{cases} 
1, & \text{if } i \in V_j; \\
0, & \text{otherwise}.
\end{cases}
\]
We shall sometimes refer to the FR code \( C \) by simply referring to its incidence matrix \( N \). In a similar manner,
we can associate a bipartite graph with the FR code \( C \). The storage nodes \( V \) in the FR code correspond to one
subset of vertices in the bipartite graph and the symbols \( \Omega \) correspond to the other disjoint subset. An edge exists
between two vertices if the corresponding entry in the incidence matrix \( N \) is non-zero.

For a FR code with incidence matrix \( N \), the transposed FR code refers to the code specified by \( N^T \). In the
transposed code, the roles of the storage nodes and the symbols are reversed.

Incidence matrices with appropriate parameters can be combined via operations such as the Kronecker product
to obtain new matrices (equivalently FR codes) with a new set of parameters. We use this technique extensively in
the sequel to generate families of FR codes.

**Definition 7:** If \( A \) is an \( m \)-by-\( r \) matrix and \( B \) is a \( p \)-by-\( q \) matrix, then the Kronecker product \( A \otimes B \) is the
\( mp \)-by-\( rq \) block matrix
\[
\begin{pmatrix}
a_{11}B & a_{12}B & \cdots & a_{1r}B \\
a_{21}B & a_{22}B & \cdots & a_{2r}B \\
\vdots & \vdots & \ddots & \vdots \\
a_{m1}B & a_{m2}B & \cdots & a_{mr}B
\end{pmatrix}.
\]
Let \( N_1 \) and \( N_2 \) be two incidence matrices of FR codes \( C_1 = (\Omega_1, V_1) \) and \( C_2 = (\Omega_2, V_2) \) with parameters
\( (n_1, \theta_1, \alpha_1, \rho_1) \) and \( (n_2, \theta_2, \alpha_2, \rho_2) \) respectively. Let \( c_1, \cdots, c_{n_1} \) be the \( n_1 \) columns of \( N_1 \) and \( d_1, \cdots, d_{n_2} \) be the
\( n_2 \) columns \( N_2 \). A new FR code can be obtained from the old one by the following incidence matrix
\[
\tilde{N} = \begin{bmatrix}
N_1 \otimes d_1 & N_1 \otimes d_2 & \cdots & N_1 \otimes d_{n_2}
\end{bmatrix}.
\]
It can be seen that \( \tilde{N} \) is equivalent to the Kronecker product of \( N_1 \) and \( N_2 \) upon appropriate permutation of its
rows and columns.

We can obtain a DSS by replicating the symbols of another DSS via Kronecker product as follows. Let \( N \) be
incidence matrix of a FR code with parameters \( (n, \theta, \alpha, \rho) \) with \( \beta = 1 \). Let \( 1 \) be the \( m \times 1 \) all-ones column vector.
The FR code obtained from \( \tilde{N} = N \otimes 1 \) has parameters \( (n, \theta m, \alpha m, \rho) \). This corresponds to the case of trivially
obtaining codes with \( \beta > 1 \) by treating every symbol in \( N \) to be consisting of \( m \) new symbols. In the subsequent
discussion we will refer to this technique for obtaining codes with \( \beta > 1 \) as trivial \( \beta \)-expansion.

In the remainder of this section, we discuss some illustrative examples of FR codes. Our first example is a code
with \( \beta > 1 \) that cannot be obtained by trivial \( \beta \)-expansion.

**Example 2:** *Non-trivial code with \( \beta > 1 \).* Consider the DSS shown in Fig. 4. The ten symbols are obtained by
using an outer \((10, 6)\) MDS code followed by the FR code illustrated in Fig. 4. Note that the DSS can recover
from a single node failure by downloading two packets each from two nodes in the same column; hence \( d = 2 \).
Moreover, it can be observed that any two nodes recover at least 6 symbols, thus \( k = 2 \). According to the Singleton
bound \( d_{\min} \leq 15 - \lceil \frac{6}{4} \rceil + 1 = 14 \). It can be seen that the code is resilient up to 13 failures (since \( k = 2 \)) and thus
meets the Singleton bound. However, this code (with \( \beta = 2 \)) cannot be arrived at simply by replication. To see this
we note that if this were true, the original DSS with \( \beta = 1 \) must correspond to a storage capacity of 2 and have a
Fig. 4. The figure shows a DSS where \( n = 15, k = 4, d = 2, \theta = 10, \alpha = 4, \rho = 6 \). A node can be repaired by contacting the other two nodes in the same column. The system is resilient up to 5 node failures.

Fig. 5. A failed node can be recovered by contacting 2 nodes and downloading 1 packet from each. The code is resilient up to 5 failures and the file size is 5. The minimum distance is of the code is 6, since any 4 nodes can recover the file.

number of symbols which is 5. However, this means that there can be at most \( \binom{5}{2} = 10 \) distinct storage nodes of capacity two. Thus our design with \( n = 15 \) cannot be obtained this way.

Next we demonstrate an example of a locally recoverable DSS, i.e., a system where \( d < k \) that is constructed using the Kronecker product method.

Example 3: Locally recoverable code using Kronecker product technique. Let \( C = (\Omega, V) \) be a FR code with \( \Omega = \{1, 2, 3\} \) and \( V = \{V_1 = \{1, 2\}, V_2 = \{2, 3\}, V_3 = \{1, 3\}\} \) with incidence matrix \( N \). The code obtained from \( \bar{N} = I \otimes N \) is presented in Fig. 5 where \( I \) denotes the \( 3 \times 3 \) identity matrix. Suppose that the outer MDS code has parameters \((9, 5)\), so that \( \theta = 9, M = 5 \). It can be observed in this construction that the file can be recovered by contacting any four nodes, so that \( k = 4 \) and that a failed node can be recovered by contacting two nodes and downloading one packet from each of them. Thus, \( d = 2 < k \). This implies that the code is locally recoverable. It can be seen that the code is resilient up to 5 failures and thus it meets the minimum distance bound in Lemma 2.

A. Discussion of related work

The work of Dimakis et al. [2] initiated the work on regenerating codes, by demonstrating the tradeoff between the storage capacity of nodes and the repair bandwidth. Their work considered functional repair, where the new node is functionally equivalent to the failed node and demonstrated that random network coding suffices for achieving this tradeoff. Following this, several papers [10], [9], [11], [12], [13], [14], [15], [7], [16] considered the construction of exact repair regenerating codes, where the new node is an exact copy of the failed node. In most cases, these constructions either operate at the minimum storage regenerating (MSR) point [10], [14], [17], [15], [11] or the minimum bandwidth regenerating (MBR) point [10], [9], [18], [7], [16]. More recently, codes with local repair
have been investigated where the metric for repair is the number of surviving nodes that are contacted for repair [3], [5], [4], [19], [20], [21].

Constructions of repair-by-transfer codes, where node repair is performed simply by downloading symbols from surviving nodes was first presented in the work of [18] where they constructed a repair-by-transfer MBR code with \( d = n - 1 \). Repair by transfer codes have also appeared in [22], [23]. The work of [7] also considered such codes (termed “exact and uncoded repair”) but with a repair degree that can be strictly smaller than \( n - 1 \). The repair operates by contacting a specific set of \( d \) surviving nodes and is hence table based. Reference [7] introduced the system architecture whereby an MDS code is applied to a file consisting of \( M \) symbols to obtain \( \theta \) symbols. These symbols are then placed onto the storage nodes and this placement is referred to as the fractional repetition code. The codes in [7], were derived from Steiner systems. They provided lower and upper bounds on the corresponding file sizes. Following this, the work of [24] constructed FR codes from bipartite cages. These codes enjoy the property that the node storage capacity is much larger than the replication degree. For the given parameters they arrive at codes with the smallest number of storage nodes. In [24], they used MOLS to construct bipartite cages and codes thus obtained are different from ours. In our construction we obtain the storage nodes directly from the set of MOLS and also obtain net FR codes. Reference [25] presents necessary and sufficient conditions on the existence of an FR code with certain parameters; however, it does not consider the issue of determining the file size for a given \( k \).

The work of [26] presents several FR code constructions based on combinatorial structures including regular and biregular graphs, graphs with a given girth, transversal designs, projective planes and generalized polygons. They consider codes where \( \alpha = d \geq k \) and \( \beta = 1 \) and show that the file size of their constructions meets the upper bound presented in [7] for \( k \leq d \). This work is closely related to the content of Section III of our work. Their construction of FR codes from transversal designs treats the blocks of the transversal design as symbols. Thus, it can be thus considered as working with the transpose of the incidence matrix corresponding to the original transversal design. Our FR codes in Section III are obtained from nets which can also be viewed as transposes of transversal designs. However, as discussed in Section III-C, the analysis of file size for our constructions cannot be obtained from the results in [26]. Our work differs in the sense that we present constructions with non-trivial \( \beta \) values, Kronecker product constructions and local FR codes.

The problem of local repair for scalar codes (\( \alpha = 1 \)) was first considered in [3]. This was extended to vector codes (\( \alpha > 1 \)) in [19], [4]. References [19], [4] study the tradeoff between locality and minimum distance and corresponding code constructions. In [19], the authors presented constructions that use local repair-by-transfer MBR codes where \( d = n - 1 \). Local codes were also studied in [27] where the design consists of an outer Gabidulin encoder followed by inner local MBR encoders. This work (see Construction III.1 in [27]) also provides examples of local FR codes by using \( t \)-designs. However, the achievable parameters are limited as \( k \) needs to be chosen to be at most \( t \) and explicit constructions of \( t \)-designs for large \( t \) are largely unknown. In Section V we focus on regenerating codes that allow a repair process in a local manner by simply downloading packets from the surviving nodes. We provide an upper bound for the minimum distance and constructions of codes which meet this bound. Our constructions use local FR codes instead of repair by transfer MBR codes. We also note that our codes are quite different from those that appear in [27], [19] and allow for a larger range of code parameters. Regenerating codes using \( t \)-designs were also presented in [12]. The architecture of the codes consists of a layered erasure correction structure that ensures a simple decoding process. These codes are showed to be achieve performance better than
time-sharing between MBR and MSR points.

III. CONSTRUCTION OF FR CODES WHEN \( k \leq d \)

Our first set of constructions are FR codes that are derived from balanced incomplete block designs (BIBDs). Following this, we construct a large family of FR codes that are derived from resolvable designs. For all these codes, the repair degree \( d \geq k \). It will be evident that resolvable designs allow us to construct FR codes where the repetition degree \( \rho \) can be varied quite easily. Furthermore, we can also obtain FR codes with normalized repair bandwidth \( \beta \) that cannot be obtained by replicating symbols. Our constructions meet the Singleton bound for specific choices of file size. We note here that constructions of FR codes based on Steiner systems with \( t = 2 \) (that are BIBDs) have appeared in the literature \[7\]. However, to our best knowledge, prior work does not provide results on the file size of the constructions. In the discussion below, we present a certain class of Steiner systems for which we can determine the file size of the FR codes obtained from their transpose.

A. FR codes from Steiner systems

We consider Steiner systems \( S(2, \alpha, \theta) \). Note that the repetition degree of any symbol is \( \rho = \frac{\alpha - 1}{\alpha - 1} \) and any two distinct symbols are contained in exactly one node. Consider the FR code \( C = (\Omega, V) \) obtained from it and its transpose.

In general, it is a challenging task to find the file size for a given FR code. For codes obtained from Steiner systems and their transposes, lower bounds based on the inclusion-exclusion principle were presented in \[7\]. However, it is important to note that the file size depends critically on the structure of the Steiner system, i.e., two different Steiner systems with the same parameters can have different file sizes. To see this, consider two non-isomorphic Steiner systems \( S(2, 3, 15) \) denoted \( D_1 \) and \( D_2 \); the nodes of these designs are provided in Tables I and II. These designs can be also found in \[28\].

Let \( S \) be a subset of symbols of the design such that no 3-subset of \( S \) is contained in a node. It can be verified by inspection that the maximum size of \( S \) in \( D_1 \) and \( D_2 \) equals 6 (\( S = \{0, 1, 3, 6, 7, 9\} \)) and 8 (\( S = \{1, 2, 4, 6, 7, 8, 9, 13\} \)) respectively. This observation results in different file sizes in the codes obtained from the transposes of \( D_1 \) and \( D_2 \), denoted \( D_1^T \) and \( D_2^T \) respectively. In fact for \( k = 7 \), the design \( D_2^T \) yields a code which has file size \( M_2 = 28 \) which matches the inclusion-exclusion lower bound given by \( 7 \times 7 - \binom{7}{2} \). However, the design \( D_1^T \) yields a code with file size \( M_1 = 29 \) which is strictly larger\[1\].

In light of the above example, it becomes imperative to consider Steiner systems that have certain restrictions. Suppose that there is a subset \( S \subset \Omega \) with \( |S| = s \) such that for each node \( V_i \in V \) we have the property that either \( |V_i \cap S| = 0 \) or \( |V_i \cap S| = 2 \). Thus, any three symbols from \( S \) are not contained in any node in \( V \). The largest set \( S \) with this property is called a maximal arc of the Steiner system \[29\]. It turns out that we can determine the file size for FR codes obtained from transposes of Steiner systems with nontrivial maximal arcs.

For a maximal arc \( S \), consider a symbol \( q \in S \). In this case there are \( s - 1 \) pairs of symbols \( (p, q) \) such that \( p, q \in S \). Since \( C \) is a Steiner system and \( S \) is a maximal arc there are \( s - 1 \) distinct nodes in \( V \) where each of these pairs occurs. Now, the repetition degree of the system is \( \rho \). Thus, there are \( \rho - (s - 1) \) nodes which contain the symbol \( q \) but no other symbol from \( S \). Based on our assumption, each node in \( V_i \in V \) is such that either \( |V_i \cap S| = 0 \) or \( |V_i \cap S| = 2 \). Thus, it has to be the case that \( s = \rho + 1 \).

\[1\]This example corrects an error in Lemma 11 of \[7\].
TABLE I
Nodes of the Steiner system \(\mathcal{D}_1\)

| \{0, 1, 2\} | \{0, 3, 4\} | \{0, 5, 6\} | \{0, 8, 7\} | \{0, 9, 10\} |
| \{0, 11, 12\} | \{0, 13, 14\} | \{1, 3, 5\} | \{1, 4, 7\} | \{8, 1, 6\} |
| \{1, 11, 9\} | \{1, 10, 13\} | \{1, 12, 14\} | \{9, 2, 3\} | \{2, 4, 6\} |
| \{2, 10, 5\} | \{2, 14, 7\} | \{8, 2, 12\} | \{2, 11, 13\} | \{3, 11, 6\} |
| \{3, 12, 7\} | \{8, 3, 13\} | \{10, 3, 14\} | \{4, 5, 13\} | \{8, 9, 4\} |
| \{4, 10, 12\} | \{11, 4, 14\} | \{11, 5, 7\} | \{8, 5, 14\} | \{9, 12, 5\} |
| \{10, 6, 7\} | \{9, 6, 14\} | \{12, 13, 6\} | \{9, 13, 7\} | \{8, 10, 11\} |

TABLE II
Nodes of the Steiner system \(\mathcal{D}_2\)

| \{1, 11, 6\} | \{1, 2, 5\} | \{2, 3, 6\} | \{9, 5, 6\} | \{3, 11, 5\} |
| \{7, 13, 5\} | \{11, 4, 13\} | \{2, 12, 7\} | \{3, 4, 7\} | \{8, 4, 5\} |
| \{8, 11, 7\} | \{4, 12, 6\} | \{8, 6, 14\} | \{12, 5, 14\} | \{8, 3, 13\} |
| \{8, 9, 12\} | \{9, 10, 13\} | \{12, 13\} | \{9, 7\} | \{9, 2, 11\} |
| \{0, 8, 2\} | \{9, 4, 14\} | \{10, 11, 14\} | \{0, 11, 12\} | \{0, 3, 14\} |
| \{8, 1, 10\} | \{10, 3, 12\} | \{1, 3, 9\} | \{0, 10, 5\} | \{0, 13, 6\} |
| \{1, 14, 7\} | \{2, 4, 10\} | \{2, 13, 14\} | \{0, 1, 4\} | \{10, 6, 7\} |

Lemma 3: Let \(\mathcal{C} = (\Omega, V)\) be a FR code derived from a Steiner system \(S(2, \alpha, \theta)\) with \(\rho = \frac{\theta - 1}{\alpha - 1}\), such that it has a maximal arc of size \(\rho + 1\). Then, the transposed FR code \(\mathcal{C}^T\) is such that its code rate is \(\frac{k\rho - \binom{k}{2}}{n\alpha}\) for \(1 \leq k \leq \rho + 1\).

Proof: In the transposed code \(\mathcal{C}^T\), consider any subset of nodes of size \(k\), where \(1 \leq k \leq \rho + 1\). As any two symbols in the original code \(\mathcal{C}\) occur in exactly one node of \(\mathcal{C}\) it holds that two nodes \(V_1\) and \(V_2\) in \(\mathcal{C}^T\) are such that \(|V_1 \cap V_2| = 1\). In addition, the storage capacity of the nodes in \(\mathcal{C}^T\) is equal to \(\rho\).

Using a simple inclusion-exclusion bound, it can be observed that these nodes cover at least \(k\rho - \binom{k}{2}\) symbols in \(\mathcal{C}^T\). Now we pick a set of \(k\) nodes in \(\mathcal{C}^T\) that correspond to a subset of the maximal arc \(S\) in \(\mathcal{C}\). Based on the argument above, it is clear that any two of these nodes intersect in exactly one symbol and any \(l\) of the nodes have an empty intersection if \(l \geq 3\). It follows that the union of these nodes has exactly \(k\rho - \binom{k}{2}\) symbols. The results follows.

The following result concerns the existence of maximal arcs in Steiner systems.

Theorem 1 (cf. Theorem 5 of [30]): Let \(\rho = \frac{\theta - 1}{3}\) be a prime power. Then, there exists an Steiner system \(S(2, 4, \theta)\) with a maximal arc of size \(\rho + 1\).

Next we provide an explicit example. Let \(\mathcal{C}\) be the FR code obtained from a Steiner system \(S(2, \alpha = 4, \theta = 16)\). The nodes in \(\mathcal{C}\) are specified in Table \(\text{III}\) and the nodes of the transposed code \(\mathcal{C}^T\) are specified in Table \(\text{IV}\).

It can be verified that the set \(S = \{0, 1, 2, 3, 4, 15\}\) is a maximal arc for this Steiner system.

According to Lemma 3, the file size for \(\mathcal{C}^T\) for \(1 \leq k \leq 6\) can be determined by just considering the nodes \(\{1, 4, 12, 13, 15\}, \{0, 2, 13, 14, 16\}, \{1, 3, 10, 14, 17\}, \{2, 4, 10, 11, 18\}, \{0, 3, 11, 12, 19\}, \text{ and } \{15, 16, 17, 18, 19\}\), which correspond to the symbols of \(S\) in \(\mathcal{C}\). For these values of \(k\), the file size of the code is \(5k - \binom{k}{2}\). Moreover, it is optimal with respect to Singleton bound for \(1 \leq k \leq 3\) (cf. Observation 1).
B. FR codes from resolvable designs

We now consider a large family of FR codes that enjoy the property of being resolvable. This allows us to obtain DSS where the repetition degree of the symbols of the FR code can be easily varied.

**Definition 8: Resolvable FR code.** Let \( C = (\Omega, V) \) where \( V = \{V_1, \ldots, V_n\} \) be a FR code. A subset \( P \subset V \) is said to be a parallel class if for \( V_i \in P \) and \( V_j \in P \) with \( i \neq j \) we have \( V_i \cap V_j = \emptyset \) and \( \bigcup_{j:V_j \in P} V_j = \Omega \). A partition of \( V \) into \( r \) parallel classes is called a resolution. If there exists at least one resolution then the code is called a resolvable FR code.

For a resolvable FR code, we call two storage nodes parallel if they belong to the same parallel class and non-parallel otherwise. The properties of a resolvable FR code are best illustrated by means of the following example.

**Example 4:** Consider a DSS with parameters \( \alpha = 3, \theta = \alpha^2 = 9, \rho = 2 \) and \( \beta = 1 \). Suppose that we arrange the symbols in \( \Omega = \{1, \ldots, 9\} \) in a \( \alpha \times \alpha \) array \( A \) shown below.

\[
A = \begin{bmatrix}
1 & 2 & 3 \\
4 & 5 & 6 \\
7 & 8 & 9 \\
\end{bmatrix}
\]

Let the rows and the columns of \( A \) form the nodes in the FR code \( C \) (see Fig. [6]), thus \( n = 6 \). It is evident that there are two parallel classes in \( C \), \( P^r = \{V_1, V_2, V_3\} \) (corresponding to rows) and \( P^c = \{V_4, V_5, V_6\} \) (corresponding to columns). As \( \rho = 2 \), this code can tolerate one failure. It can be observed that for \( V_i \in P^r \) and \( V_j \in P^c \), we have \( |V_i \cap V_j| = 1 \). Using this we can compute the file size \( M \) when \( k = 3 \), as follows. Let \( a + b = 3 \) with \( a \geq b \). Then, the number of distinct symbols in a set of 3 nodes from \( C \) is

\[
3a + (3-a)(3-a) = a^2 + 9 - 3a,
\]

where \( a \) nodes are from \( P^r \) and \((3-a)\) nodes are from \( P^c \). This is minimized when \( a = 2 \). Thus, \( M = 7 \) and \( R_C = \frac{7}{18} \). Note also that the code is optimal with respect to the Singleton bound.
Fig. 6. A DSS specified with \((n = 6, k = 3, d = 3, \alpha = 3)\). Note that \(\{V_1, V_2, V_3\}\) and \(\{V_4, V_5, V_6\}\) form parallel classes.

It can be seen that if one starts with a resolvable design with many parallel classes, the repetition degree \(\rho\) can be varied easily by adding and/or removing parallel classes if needed. We emphasize that the constructions of [7] that are based on Steiner systems largely lack this flexibility as many of them are not resolvable.

In our proposed systems, we require recovery from a node failure by downloading exactly \(\beta\) symbols each from a specified set of \(d\) surviving nodes. In the case of \(\beta = 1\) we just need to ensure that each coded symbol is repeated \(\rho > 1\) times to address this requirement. However the case of \(\beta > 1\) is no longer as simple; we need to guarantee the existence of a set \(S_{V_i}\) of size \(d\) such that for each failed node \(V_i\), \(V_i\) must be \(\beta\)-recoverable from \(S_{V_i}\). To address this issue, we consider a subclass of resolvable FR codes called net FR codes where the intersection size of any two nodes from distinct parallel classes is exactly \(\beta\).

**Definition 9: Net FR code.** Let \(C = (\Omega, V)\) be a resolvable FR code with parameters \((n = ar, \theta = a^2b, \alpha = ab, \rho = r)\) such that any two non-parallel nodes intersect in exactly \(b\) symbols. The incidence structure determined by \(C\) is called a net [29] and we call \(C\) a net FR code.

Examples of net FR codes can be obtained from several combinatorial structures, e.g., grids, affine resolvable designs, Hadamard designs and mutually orthogonal Latin squares (MOLS). We elaborate on these constructions in the subsequent discussion.

Suppose that a net FR code with parameters \((n = ar, \theta = a^2b, \alpha = ab, \rho = r)\) exists. Note that the number of nodes in a parallel class equals \(\theta / \alpha = a\). Furthermore if a given node \(V_1 \in V\) fails, it can be observed that this node can be reconstructed by contacting all the nodes in any other intact parallel class and downloading \(b\) symbols from each of them. This implies that the code has \(d = a, \beta = b\). Next, the code has \(\rho = r\) parallel classes and any node can be reconstructed as long as there exists at least one parallel class. Thus, the code is resilient to at least \(r - 1\) failures, i.e., \(\rho_{res} = r - 1\).

Note that the parameter \(k\) can be chosen such that \(1 \leq k \leq d\). The code rate \(R_C\) depends on \(k\) and needs to be determined. As we shall see determining \(R_C\) can be nontrivial in many cases. Specifically, much of the literature in the area of combinatorial designs focuses on pairwise intersections between the storage nodes, whereas the code rate \(R\) depends on the minimum size of the intersection of any \(k\) storage nodes. Some general results about the code rate of net FR codes can be obtained as discussed in the lemma below. However, a more careful analysis of the algebraic structure of a given construction can allow us to arrive at stronger results.

**Lemma 4:** Let \(C\) be a net FR code with parameters \((n = ar, \theta = a^2b, \alpha = ab, \rho = r)\) such that \(\beta = 1\). Let \(k\) be an integer that satisfies \(k \leq \rho\) and \(\binom{k-1}{2} < a\). Then, the code rate of the system is \(R_C = \frac{(\alpha k - \binom{k}{2})}{n\alpha}\).

**Proof:** See Appendix.

**Example 5:** Consider the following FR code obtained from a net with parameters \((n, \theta, \alpha, \rho) = (16, 16, 4, 4)\). This FR code can be specified the nodes presented in Table V. Each row of the table represents a parallel class.
It can be observed that any two non-parallel nodes intersect in exactly one point. Thus, the code corresponds to a net FR code with $a = 4, b = 1$, and $r = 4$. Thus, $d = 4$ and $\beta = 1$. Suppose that $k = 4$, so that $\binom{k-1}{2} < a$. Our algorithm (cf. Appendix) may choose the following nodes for $k = 4$.

$$L = \{\{1, 2, 3, 4\}, \{1, 5, 9, 13\}, \{2, 5, 12, 15\}, \{4, 6, 9, 15\}\}.$$  

So the file size is $\mathcal{M} = 10$. However this code is not optimal with respect to Singleton bound. It can be observed that the code formed by deleting a parallel class has parameters $(n = 12, \theta = 16, \alpha = 4, \rho = 3)$. In this code any three nodes cover at least 9 symbols. Thus setting $k = \lceil \frac{n}{q} \rceil = 3$ (cf. Observation 1) results in a code that meets the Singleton bound.

Note that while Lemma 4 applies to all net FR codes, the requirement that the storage capacity $\alpha = a \geq \binom{k-1}{2}$ is quite restrictive. For certain net FR codes that have a tractable algebraic and/or geometric characterization we can perform a more careful analysis and we now turn our attention to them. Our first example is a net where the file size $\mathcal{M}$ is strictly larger than $k\alpha - \beta\binom{k}{2}$.

1) **Affine Resolvable FR code:** Let $q$ be a prime power, $m \geq 2$ and $\Omega = \mathbb{F}_q^m$. Let $1 \leq \delta \leq m - 1$. We treat $\Omega$ as an $m$-dimensional vector space over $\mathbb{F}_q$. A $\delta$-flat is the solution set to a system of $m - \delta$ independent linear equations that can be homogeneous or non-homogeneous. The set $\Omega$ and the set of all $\delta$-flats of $\Omega$ comprise the $m$-dimensional affine geometry over $\mathbb{F}_q$, denoted by $AG_m(q)$. It turns out that one can generate a large class of resolvable designs by considering $AG_m(q)$. Let $\left[ m \atop \delta \right]_q$ denote the Gaussian coefficient, so that

$$\left[ m \atop \delta \right]_q = \begin{cases} \frac{(q^{m-1})(q^{m-1} - 1) \cdots (q^{\delta+1}) - 1}{(q^{\delta}) - 1(q^{\delta} - 1) \cdots (q - 1)} & \text{if } \delta \neq 0, \\ 1 & \text{if } \delta = 0. \end{cases}$$

**Theorem 2:** [8] Let $V$ denote the set of all $\delta$-flats in $AG_m(q)$. Then $\Omega = \mathbb{F}_q^m$ and $V$ form a resolvable $(q^m, \rho, q^\delta, \lambda)$-BIBD with $n = q^{m-\delta} \left[ m \atop \delta \right]_q$, $\rho = \left[ m \atop \delta \right]_q$, and $\lambda = \left[ m-1 \atop \delta-1 \right]_q$.

The case of $m = 2, \delta = 1$ corresponds to a certain combinatorial structure known under the name of affine planes. It can be shown that when $\delta = m - 1$, using Theorem 2 we obtain an affine resolvable BIBD with $n = \theta + \rho - 1$. In this case the DSS parameters are $\theta = q^m$, $\alpha = q^{m-1}$, $\rho = \frac{q^{m-1}}{q-1}$ and $n = q\rho$. The design can be specified by means of the following algorithm.

(i) Let $\Omega = \{(x_1, x_2, \cdots, x_m) : x_i \in \mathbb{F}_q \text{ for } i = 1, 2, \cdots, m\}$ be the symbol set.

(ii) Find $\rho$, $(m - 1)$-dimensional subspaces of $\mathbb{F}_q^m$ such that each of them contains the symbol $(0, 0, \cdots, 0) \in \mathbb{F}_q^m$. Note that these subspaces of $\mathbb{F}_q^m$ are the solutions to a single homogeneous linear equation over $\mathbb{F}_q$ in $q$ variables. These $\rho$ subspaces are representatives of the $\rho$ different parallel classes.
(iii) Construct each parallel class by considering the additive cosets of its representative. Let $R_1$ be a $(m - 1)$-dimensional subspace corresponding to a given homogenous equation. Let $U = \{0, u_1, \ldots, u_{q-1}\}$ be the full set of coset representatives of $R_1$. Rest of the blocks can be obtained by the cosets $R_i^i = u_i + R_1$. Note that each of these cosets corresponds to a nonhomogeneous equation.

**Example 6:** [8] Let $q = 3$ and $m = 3$. The set of symbols is $\Omega = \mathbb{F}_3^2$ and there are 39 blocks which can be partitioned into 13 parallel classes. The representatives of the 13 parallel classes are specified in the Table VI where the vector $(x_1, x_2, x_3)$ is simply written as $x_1x_2x_3$. The other blocks are additive cosets of these 13 representatives. For example, the first parallel class consist of the following blocks:

\[
R_1 = \{000, 001, 002, 010, 020, 011, 012, 021, 022\},
\]
\[
R_2 = \{000, 001, 002, 100, 200, 101, 102, 201, 202\},
\]
\[
R_3 = \{000, 001, 002, 110, 220, 111, 112, 221, 222\},
\]
\[
R_4 = \{000, 001, 002, 120, 210, 121, 122, 211, 212\},
\]
\[
R_5 = \{000, 010, 020, 100, 200, 110, 120, 210, 220\},
\]
\[
R_6 = \{000, 010, 020, 101, 201, 102, 112, 212, 222\},
\]
\[
R_7 = \{000, 010, 020, 101, 201, 112, 122, 211, 221\},
\]
\[
R_8 = \{000, 011, 022, 100, 200, 111, 112, 211, 222\},
\]
\[
R_9 = \{000, 011, 022, 101, 201, 112, 122, 212, 222\},
\]
\[
R_{10} = \{000, 011, 022, 102, 201, 110, 121, 212, 220\},
\]
\[
R_{11} = \{000, 012, 021, 100, 200, 112, 121, 212, 221\},
\]
\[
R_{12} = \{000, 012, 021, 101, 202, 110, 122, 211, 220\},
\]
\[
R_{13} = \{000, 012, 021, 102, 201, 111, 120, 210, 222\}.
\]

Here the blocks $B_1$, $B_2$ and $B_3$ correspond to equations $x_1 = 0$, $x_1 = 1$ and $x_1 = 2$ respectively.

The overlap between blocks from different parallel classes in the case of affine resolvable designs is known from the following result.

**Lemma 5:** [8] Any two blocks from different parallel classes of an affine resolvable $(\theta, \rho, \alpha, \lambda)$-BIBD intersect in exactly $\alpha^2/\theta$ symbols.

Using the above facts, we can conclude that an affine resolvable BIBD is an instantiation of a net FR code with parameters $(n = q^m-q^{-1}, \theta = q^m, \alpha = q^m-1, \rho = \frac{q^m-1}{q-1})$ and $d = q, \beta = q^m-2$. Of course, the repetition degree can be varied by only retaining as many parallel classes as needed.

**Remark 1:** It is important to note that the affine resolvable FR codes are an example of a FR code family with $\beta > 1$ that cannot be obtained by replicating the symbols of a smaller code. Specifically, consider $m \geq 2q + 1$ and $q \geq 3$. In this case the affine resolvable FR code will have parameters $\theta = q^m, \alpha = q^m-1, \rho = \frac{q^m-1}{q-1}, n = q\rho$ and $\beta = q^m - 2$. If it could be generated from a smaller code simply by replication, this would imply that the smaller
code had a storage capacity of \( q \) and \( q^2 \) total symbols. This means it has at most \( \left( \frac{q^2}{q^2} \right)^q \leq \left( \frac{q^2}{q} \right)^q = (eq)^q \leq q^{2q} \) distinct storage nodes. However, in the affine resolvable FR code we have \( n = q^{m-1} - q^{2g+1-1} \) which can be verified to be strictly larger than \( q^{2q} \).

We can determine the file size of a code \( C \) obtained from some specific affine resolvable designs, for certain ranges of \( k \). We consider two scenarios depending on the relationship between \( q \) and \( m \).

- **Case 1: \( q > m \)**

  We choose the code \( C \) such that it has \( r \geq m \) parallel classes such that the \( i \)-th parallel class of \( C \) corresponds to the homogeneous equation \( x_1 + \alpha_1 x_2 + \alpha_2^2 x_3 + \cdots + \alpha_{i-1} x_m = 0 \), where \( \alpha_i, i = 1, \ldots r \) are all non-zero and distinct. Note that the distinctness requirement also enforces that \( q > r \). It can be observed that the equations obtained in this manner are such that any \( m \) equations are linearly independent [31].

  For this code we analyze the file size for a fixed \( k \leq m \). For a given set of \( k \) blocks, denoted \( A_i, i = 1, \ldots k \), it is possible that multiple blocks from the same parallel class are chosen; suppose that these blocks come from \( l \) distinct parallel classes, numbered without loss of generality as \( 1, \ldots, l \). Let \( z_i \) denote the number of blocks from the \( i \)-th parallel class, so that

  \[
  z_1 + z_2 + \cdots + z_l = k.
  \]

  If we pick \( k_1 \) blocks each from a different parallel class, we can immediately conclude that the total number of symbols covered is \( q^{m-k_1} \), as the parallel classes correspond to linearly independent equations. Using this fact and the inclusion-exclusion principle, we have

  \[
  |\bigcup_{i=1}^k A_i| = \sum_{i_1=1}^l z_{i_1} q^{m-1} - \sum_{i_1 < i_2} z_{i_1} z_{i_2} q^{m-2} + \sum_{i_1 < i_2 < i_3} z_{i_1} z_{i_2} z_{i_3} q^{m-3} + \cdots + (-1)^l z_1 z_2 \cdots z_l q^{m-l}.
  \]

  Upon inspection, it can be observed that

  \[
  q^m \left( 1 - \Pi_{i=1}^l \left( 1 - \frac{z_i}{q} \right) \right) = \sum_{i_1=1}^l z_{i_1} q^{m-1} - \sum_{i_1 < i_2} z_{i_1} z_{i_2} q^{m-2} + \sum_{i_1 < i_2 < i_3} z_{i_1} z_{i_2} z_{i_3} q^{m-3} + \cdots + (-1)^l z_1 z_2 \cdots z_l q^{m-l}.
  \]

  Thus, we need to analyze the minimum value of the LHS of equation (3) (over the possibilities for \( z_i, i = 1, \ldots l \)) to determine the file size. Using the AM-GM inequality, it can be seen that

  \[
  \frac{1}{l} \sum_{i=1}^l \left( 1 - \frac{z_i}{q} \right) = 1 - \frac{k}{lq} \geq \left[ \Pi_{i=1}^l \left( 1 - \frac{z_i}{q} \right) \right]^{\frac{1}{l}}
  \]

  \[
  \Rightarrow \left[ 1 - \frac{k}{lq} \right]^{\frac{1}{l}} \geq \Pi_{i=1}^l \left( 1 - \frac{z_i}{q} \right).
  \]

  Equality holds in the above equation when all the \( z_i \) terms are equal. In addition we show below that the function

  \[
  h(l) = \left[ 1 - \frac{k}{lq} \right]^{\frac{l}{l}}
  \]

  takes its maximum value over the set \( l = 1, \ldots, k \) when \( l = k \). To see this, let \( 0 < \chi = \frac{k}{q} < 1 \), and consider \( \log h(l) = l \log (1 - \frac{k}{l}) \). Now,

  \[
  \frac{d}{dl} \log h(l) = \log(1 - \frac{\chi}{l}) + \frac{\chi}{1 - \frac{\chi}{l}}.
  \]
Let $\chi_1 = \frac{1}{\lambda}$ and let us study the function $h_1(\chi_1) = \log(1 - \chi_1) + \frac{\chi_1}{1 - \chi_1}$. Clearly $h_1(0) = 0$. The derivative of $h_1(\chi_1)$ is non-negative for $0 < \chi_1 < 1$, since it equals $\frac{\chi_1}{(1 - \chi_1)^2}$. This implies that $h_1(\chi_1) \geq 0$ for $0 < \chi_1 < 1$ and therefore $h'(l) \geq 0$ in the range $l = 1, \ldots, k$, i.e., it is an increasing function in this range. This implies that the maximum value of $h(l)$ in the range $l = 1, \ldots, k$ is obtained when $l = k$ and $z_i = 1$ for all $i$.

We conclude that the minimum value of the LHS of equation (3) is obtained when $k = l$ and $z_i = 1, i = 1, \ldots, k$ and that the file size is $q^m \left( 1 - \left( 1 - \frac{1}{q} \right)^k \right)$.

- **Case 2: $q \leq m$**

  In this case we choose the code $C$ so that it has $r \leq m$ parallel classes. The chosen parallel classes are such that they belong to linearly independent equations. Once again, we can analyze the file size when $k \leq m$.

  Suppose that we choose $l$ parallel classes and let $z_i$ denote the number of blocks chosen from the $i$-th parallel class. Note that in this case $l \geq \lceil \frac{q}{q} \rceil$ and $z_i \leq q$ for all $i = 1, \ldots, l$. Proceeding as in Case 1, we can argue that the function

  $$h(l) = \left[ 1 - \frac{k}{lq} \right]^l$$

  attains its maximum when $l = k$ and $z_i = 1$ for all $i = 1, \ldots, k$. Thus, in this case as well the maximum file size is given by $q^m \left( 1 - \left( 1 - \frac{1}{q} \right)^k \right)$.

2) **Resolvable FR codes from Grids, Hadamard designs and MOLS:** A $a \times a$ grid is a FR code that is obtained as follows.

- Let $\Omega = \{0, \ldots, a^2 - 1\}$. Create a 2D-array $A$ whose $(i, j) - th$ entry is $a \cdot i + j$, where $0 \leq i, j \leq a - 1$.
- Each column and each row of $A$ determines a storage node.

It is clear that the FR code so obtained is resolvable. Specifically, the set of columns and the set of rows form a resolution. The parameters are $(n = 2a, \theta = a^2, \alpha = a, \rho = 2)$. Note that $\beta = 1$ as any row and any column intersect in exactly one symbol. Thus, the code so obtained is also a net FR code.

**Lemma 6:** Let $C$ be a net FR code obtained from an $a \times a$ grid. If $k$ is even, the file size $M$ of $C$ is $ka - k^2/4$ and if $k$ is odd, it is $ka - (k^2 - 1)/4$.

**Proof:** Assume we choose $s$ nodes from the parallel class corresponding to the rows and $t$ nodes from the parallel class corresponding to the columns such that $s + t = k$. Note that $k \leq d = a$. It is evident that any three nodes have an empty intersection. Thus applying the inclusion-exclusion principle, we conclude that any $k$ nodes cover exactly $\alpha k - st$ symbols. Next, note that $\alpha k - st = ak - ks + s^2 = (s - k/2)^2 + ka - k^2/4$ which takes the minimum value $ka - k^2/4 + \min((k/2 - \lfloor k/2 \rfloor)^2, (k/2 - \lfloor k/2 \rfloor)^2)$, i.e., it equals $ka - k^2/4$ when $k$ is even and $ka - (k^2 - 1)/4$ when $k$ is odd.

The following corollary can be obtained by examining conditions under which $k = \lceil \frac{M}{\alpha} \rceil$.

**Corollary 1:**

- Let $k = 2u$ and $u^2 < a$. Then the FR code obtained from $a \times a$ grid is optimal with respect to the Singleton bound.
- Let $k = 2u + 1$ and $u(u + 1) < a$. Then the FR code obtained from $a \times a$ grid is optimal with respect to the Singleton bound.

A second construction of affine resolvable designs can be obtained from Hadamard matrices or equivalently difference sets as discussed below. Consider an algebraic group $G$ of order $\theta$ and $D \subseteq G$ such that $|D| = \alpha$, with the property that every nonidentity element of $G$ can be expressed as a difference $d_1 - d_2$ of elements of $D$ in exactly $\lambda$ ways. We refer to $D$ as a $(\theta, \alpha, \lambda)$-difference set.
Lemma 7: Quadratic residue difference set. [8] Let $q = 4a - 1$ be an odd prime power and $G = \mathbb{F}_q$. Let $D = \{z^2 : z \in \mathbb{F}_q, z \neq 0\}$ be the set of quadratic residues. Then $D$ is a $(4a - 1, 2a - 1, a - 1)$-difference set in $(\mathbb{F}_q, +)^2$.

For any $g \in G$, we define the translate of $D$ by $g + D = \{g + d : d \in D\}$, and define the development of $D$ by $\text{Dev}(D) = \{g + D : g \in G\}$. If $D$ is a $(\theta, \alpha, \lambda)$-difference set in $G$, then $(G, \text{Dev}(D))$ is a $(\theta, \rho, \alpha, \lambda)$-BIBD [8].

Let $(\Omega, V)$ be the $(4a - 1, 2a - 1, 2a - 1, a - 1)$-BIBD constructed by using a quadratic residue difference set. Let $\infty \notin \Omega$, and define for $V' = \{B \cup \{\infty\} : B \in V\}$. Then it can be shown that $(\Omega \cup \{\infty\}, V' \cup \{\Omega - B : B \in V\})$ is an affine resolvable $(4a, 4a - 1, 2a, 2a - 1)$-BIBD. Using the equations (1) and (2) it can be seen that this corresponds to a net FR code with parameters $\theta = 4a, \alpha = 2a, \beta = a, d = 2, \rho = 4a - 1$ and $n = 8a - 2$.

Example 7: $D = (1, 2, 4)$ is a $(7, 3, 1)$-difference set in $\Omega = \mathbb{F}_7$. We can construct the Fano plane by using the difference set $D$ which is a $(7, 3, 3, 1)$-BIBD. By applying the above construction we can construct a FR code with parameters $\theta = 8, n = 14, \alpha = 4, \rho = 7$. Corresponding storage nodes are presented in Table VII where each row of the table represents a parallel class.

Remark 2: For this class of codes, $d$ is always 2. However, they offer more flexibility in the choice of $\beta$; unlike affine geometry based codes, we do not require $\beta$ to be a prime power. In addition, they provide another example of a family of FR codes that cannot be obtained by trivial $\beta$-expansion. For instance, if $a = 2$, we have $\theta = 8, n = 14, \alpha = 4, \rho = 7$. If such a code could be obtained by trivial $\beta$-expansion, then the original code would correspond to an FR code with 4 symbols and storage capacity of 2. In this case, it can be observed that there can be at most $\binom{4}{2} = 6$ nodes. In contrast, the code obtained from the Hadamard design has 14 nodes. It can be seen that any $k = 2$ nodes cover at least 3a symbols where $\alpha = 2a$. Moreover, $k = 2 = \lceil\frac{3\alpha}{2}\rceil$. Hence the code is optimal with respect to Singleton bound for $k = 2$.

We now discuss another construction of net FR codes that can be obtained from MOLS.

Definition 10: A Latin square of order $a$ with entries from a set $\Omega$ with $|\Omega| = a$ is an $a \times a$ array $L$ in which every cell contains an element of $\Omega$ such that every row of $L$ is a permutation of $\Omega$ and every column of $L$ is a permutation of $\Omega$.

Definition 11: Suppose that $L_1$ and $L_2$ are Latin squares of order $a$ with entries from $\Omega_1$ and $\Omega_2$ respectively (where $|\Omega_1| = |\Omega_2|$). We say that $L_1$ and $L_2$ are orthogonal Latin squares if for every $x \in \Omega_1$ and for every $y \in \Omega_2$ there is a unique cell $(i, j)$ such that $L_1(i, j) = x$ and $L_2(i, j) = y$.

\begin{table}[h]
\centering
\begin{tabular}{|c|c|}
\hline
{\infty, 1, 2, 4} & {0, 3, 5, 6} \\
{\infty, 2, 3, 5} & {1, 4, 6, 0} \\
{\infty, 3, 4, 6} & {2, 5, 0, 1} \\
{\infty, 4, 5, 0} & {3, 6, 1, 2} \\
{\infty, 5, 6, 1} & {4, 0, 2, 3} \\
{\infty, 6, 0, 2} & {5, 1, 3, 4} \\
{\infty, 0, 1, 3} & {6, 2, 4, 5} \\
\hline
\end{tabular}
\caption{Hadamard design obtained from the $(7, 3, 1)$-difference set in $\Omega = \mathbb{F}_7$.}
\end{table}
Equivalently, one can consider the superposition of $L_1$ and $L_2$ in which each cell $(i,j)$ is occupied by the pair $(L_1(i,j), L_2(i,j))$. Then, $L_1$ and $L_2$ are orthogonal if and only if the resultant array has every value in $\Omega_1 \times \Omega_2$. A set of $r$ Latin squares $L_1, \ldots, L_r$ of order $a$ are said to be mutually orthogonal if $L_i$ and $L_j$ are orthogonal for all $1 \leq i < j \leq r$.

We now demonstrate a procedure of constructing net FR codes from MOLS [32]. Let $\Omega = \{1, 2, \ldots, a^2\}$, and let $L_1, L_2, \ldots, L_{r-2}$ be a set of $r-2$ MOLS of order $a$ ($r-2 \leq a-1$).

- Arrange the elements of $\Omega$ in an $a \times a$ array $A$. Each row and each column of $A$ corresponds to a storage node (this gives us $2a$ nodes).
- Note that $L_i$ takes values in $\{1, \ldots, a\}$. Within $L_i$ identify the set of $(i,j)$ pairs where a given value $z \in \{1, \ldots, a\}$ appears. Create a storage node by including the entries of $A$ corresponding to the identified $(i,j)$ pairs.
- Repeat this for each $L_i$ and all $z \in \{1, \ldots, a\}$. This creates another $(r-2)a$ storage nodes.

Thus, a total of $ra$ storage nodes of size $a$ can be obtained. Of course one can choose fewer storage nodes if so desired.

Lemma 8: The construction procedure described above produces a net FR code with $\theta = a^2$, $n = ra$, $d = a$, $\alpha = a$, and $\rho = r$. Where non-parallel nodes intersect in exactly one point.

Proof: It is clear from the construction that $\theta = a^2$ and $n = ra$. Each storage node has $a$ symbols so that $\alpha = a$. We need to show that the code is resolvable. Towards this end note that it is evident that we obtain a parallel class by considering the nodes corresponding to the rows of $A$ (similar argument holds for the columns of $A$). Next, the nodes obtained by considering Latin square $L_i$ also form a parallel class, since the set of elements obtained by considering the $(i,j)$ pairs corresponding to $z_1 \in \{1, \ldots, a\}$ are distinct from those corresponding to $z_2 \in \{1, \ldots, a\}$, if $z_1 \neq z_2$. As we have $r$ parallel classes, we obtain $\rho = r$. Next, consider the overlap between any two storage nodes belonging to different parallel classes. As $L_i$ and $L_j$ are orthogonal, any entry $(k,l) \in [a] \times [a]$ appears exactly once in the superposition of $L_i$ and $L_j$, which implies that the overlap between storage nodes from different parallel classes corresponding to the $L_i$’s is exactly one element. Similarly, a block from a parallel class corresponding to $L_i$ has exactly one overlap with the blocks corresponding to the rows and columns of $A$.

Example 8: Let $a = 4$, and $r = 2$. Then, we have the following construction.

$$A = \begin{array}{cccc}
1 & 2 & 3 & 4 \\
5 & 6 & 7 & 8 \\
9 & 10 & 11 & 12 \\
13 & 14 & 15 & 16
\end{array},$$

$$L_1 = \begin{array}{cccc}
1 & 2 & 3 & 4 \\
2 & 1 & 4 & 3 \\
3 & 4 & 1 & 2 \\
4 & 3 & 2 & 1
\end{array} \quad \text{and} \quad L_2 = \begin{array}{cccc}
1 & 2 & 3 & 4 \\
3 & 4 & 1 & 2 \\
4 & 3 & 2 & 1 \\
2 & 1 & 4 & 3
\end{array}$$

where it can be verified that $L_1$ and $L_2$ are orthogonal. We have the parallel classes and corresponding storage nodes illustrated in Example 5.

Note that in describing the above construction we assumed the existence of $r-2$ MOLS. We now discuss the issue of the existence of such structures.
If $p$ is a prime number, $m$ is a positive integer, and $N = p^m$ then we can construct $N - 1$ mutually orthogonal Latin squares as described below.

(i) Define $L_a : \mathbb{F}_N \times \mathbb{F}_N \rightarrow \mathbb{F}_N$, by $(r, c) \mapsto ar + c$ (where the addition is over $\mathbb{F}_N$) for all $a \in \mathbb{F}_N \setminus \{0\}$. Then $L_a$ is a Latin square since for a given row $r$ (or column $c$) the column (or row) location of an element $s$ is uniquely specified.

(ii) For any $a, b \in \mathbb{F}_N \setminus \{0\}$, $L_a$ and $L_b$ are orthogonal since for given ordered pair $(s, t)$ the system $ar + c = s$, $br + c = t$, determine $r = (a - b)^{-1}(s - t)$ and $c = s - ar$ uniquely.

Example 9: Let $N=3$. Then $\mathbb{F}_3 = \{0, 1, 2\}$, $L_1 : x + y$ and $L_2 : 2x + y$. The two orthogonal Latin squares of order 3 constructed by the above method are

\[
L_1 = \begin{pmatrix} 0 & 1 & 2 \\ 1 & 2 & 0 \end{pmatrix}, \quad \text{and} \quad L_2 = \begin{pmatrix} 2 & 0 & 1 \\ 0 & 1 & 2 \end{pmatrix}
\]

Remark 3: In general, the construction of orthogonal Latin squares is somewhat involved. However, the celebrated results of [33], demonstrate the construction of two orthogonal Latin squares for all orders $N \neq 2, 6$. This immediately allows us to construct net FR codes with the following parameters $n = 4a, \theta = a^2, d = \alpha = a, \beta = 1$, and $\rho = 4$ for any $a \neq 2, 6$. By applying Lemma 4 we can get the file size $\mathcal{M} = 4a - 6$ for $k = 4$ for $a > 6$ and it is optimal with respect to Singleton bound (cf. Observation 1).

It can be observed that this construction allows us to design some FR codes that cannot be obtained from Steiner systems. For instance, Let $\alpha = 10$ and $\theta = 100$. Then to construct a FR code we need use the Steiner system $S(2, 10, 100)$ which does not exist [34]. However the above construction with two orthogonal Latin squares of order 10 provides us a net FR code with $\alpha = 10$ and $\theta = 100$.

Lemma 9: Let $p$ be a prime and $m$ be a positive integer, so that there exist $p^m - 1$ MOLS of order $p^m$. Consider a subset of these $p^m - 1$ MOLS of size $r$ and let $C$ be a net FR code constructed from them. Then for any $k \leq r$, the code rate $R_C = (k(p^m) - \binom{k}{2})/np^m$.

Proof: Let $\eta$ be a primitive element of $\mathbb{F}_{p^m}$. From the construction of the $r$ MOLS, we can associate a set of non-zero field elements $\{\eta^{\alpha_1}, \ldots, \eta^{\alpha_r}\}$ so that the $i$-th Latin square is generated by the corresponding $\eta^{\alpha_i}$, where $\alpha_i$'s are distinct. In the discussion below we demonstrate the existence of $r$ storage nodes that cover exactly $rp^m - \binom{r}{2}$ symbols. The argument will also show the required result for any $k < r$. From the inclusion-exclusion principle it is evident that any $r$ nodes cover at least $rp^m - \binom{r}{2}$ symbols. For demonstrating a set of nodes that cover exactly this number we need to ensure that the intersection of any three storage nodes from the set is empty. Note that from the previous discussion it is clear that any two nodes intersect in at most one symbol, thus we need to ensure the pairwise intersections are all distinct.

Towards this end in the $i$-th MOLS, consider the storage node determined by the equation $\eta^{\alpha_i}x + y = \eta^{2\alpha_i}$. This specifies the set of nodes that we will be considering. Three nodes intersect in some symbol if the following system of equations has a solution.

\[
\eta^{\alpha_i}x + y = \eta^{2\alpha_i} \quad (4)
\]

\[
\eta^{\alpha_j}x + y = \eta^{2\alpha_j} \quad (5)
\]

\[
\eta^{\alpha_k}x + y = \eta^{2\alpha_k} \quad (6)
\]
Note that any two equations from the set above are linearly independent and have exactly one solution. Thus, if the above system has a solution, then there exist $\mu \neq 0$ and $\lambda \neq 0$ such that

$$
\lambda \eta^{\alpha_i} + \mu \eta^{\alpha_j} = \eta^{\alpha_k}
$$

$$
\lambda + \mu = 1
$$

$$
\lambda \eta^{2\alpha_k} + \mu \eta^{2\alpha_j} = \eta^{2\alpha_k}
$$

Next, we note that it cannot be the case that $\eta^{2\alpha_i} = \eta^{2\alpha_j} = \eta^{2\alpha_k}$. To see this note that there are no zero divisors in a finite field so $z_1^2 = z_2^2$ implies $z_1 = z_2$ or $z_1 = -z_2$. Thus, we can conclude that

$$
\lambda = \frac{\eta^{\alpha_k} - \eta^{\alpha_j}}{\eta^{\alpha_i}} = \frac{\eta^{2\alpha_k} - \eta^{2\alpha_j}}{\eta^{2\alpha_i} - \eta^{2\alpha_j}}.
$$

However

$$
\frac{\eta^{2\alpha_k} - \eta^{2\alpha_j}}{\eta^{2\alpha_i} - \eta^{2\alpha_j}} = \frac{(\eta^{\alpha_k} - \eta^{\alpha_j})(\eta^{\alpha_k} + \eta^{\alpha_j})}{(\eta^{\alpha_i} - \eta^{\alpha_j})(\eta^{\alpha_i} + \eta^{\alpha_j})}
$$

and this implies $\eta^{\alpha_i} = \eta^{\alpha_k}$ which is a contradiction. Thus, a solution to the system of equations in (4) - (6) does not exist. The result follows.

Remark 4: The existence of $p^m - 1$ MOLS implies the existence of an affine plane of order $p$ [8]. Thus choosing $r = p^m - 1 = k$, we can obtain the corresponding file sizes for affine planes. Codes constructed from affine planes were also considered in [8] under Steiner systems.

Example 10: An FR code obtained from affine plane of order 3 is depicted in Fig. 7. This code can be obtained by following the construction outlined above with $p = 3$ and $m = 1$. It can be observed that this code is optimal with respect to the Singleton bound when $k = 2$. (cf. Observation 1).

C. Discussion of code parameters achieved by the proposed constructions

In this subsection, we summarize the range of DSS parameters that our constructions can achieve. Note that there are certain parameter restrictions that any FR code has to satisfy. We list these below. To avoid trivialities, we assume there are no repeated storage nodes in the system.

$$
n \alpha = \theta \rho, \text{ by counting the number of ones in the incidence matrix},
$$

$$
n \leq \binom{\theta}{\alpha}, \text{ as the nodes are } \alpha\text{-sized subsets of the symbols},
$$

$$
2 \leq \alpha \leq \theta - 1, \text{ as the storage capacity can be at most } \theta - 1,
$$
TABLE VIII

CONSTRUCTIONS WHERE $\beta = 1$. NOTE THAT WE CAN PERFORM TRIVIAL $\beta$-EXPANSION TO OBTAIN HIGHER $\beta$.

| Method | $(n, \theta, \alpha, \rho)$ | $\mathcal{M}$ | Range of $k$ | Comments |
|--------|----------------------|----------------|-------------|----------|
| Steiner Systems | $(n, \theta, \alpha, \frac{\theta-1}{\alpha-1})$ | $\geq k\alpha - \binom{k}{2}$ | $1 \leq k \leq \alpha$ | Known to exist for certain specific parameters [8]. |
| Transposed Steiner Systems | $(n, \theta, \frac{\theta-1}{\rho-1}, \rho)$ | $\geq k\alpha - \binom{k}{2}$ | $1 \leq k \leq \alpha$ | Equality in the file size lower bound is guaranteed if the original Steiner system has a maximal arc. There are large families of Steiner systems where maximal arc exist [30]. |
| Grids | $(2a, a^2, a, 2)$ | $ka - k^2/4$ for even $k$, $ka - (k^2-1)/4$ for odd $k$ | $1 \leq k \leq a$. | $a$ can be a composite number. |
| MOLS (Remark 3) | $(4a, a^2, a, 4)$ | $\geq k\alpha - \binom{k}{2}$ | $1 \leq k \leq 4$ | Equality in the file size lower bound holds if $k = 4$. Use the construction of two MOLS for order greater than 6 [33]. |
| MOLS (Lemma 9) | $(\rho p^m, p^{2m}, p^m, \rho \leq p^m - 1)$ | $kp^m - \binom{k}{2}$ | $1 \leq k \leq \rho$ | $\rho$ is a prime. Use the construction of MOLS where the order is a prime power. |

We emphasize that any FR code is equivalent to a biregular bipartite graph and the file size for a given value of $k$ is closely related to the expansion properties of $k$-sized subsets of the storage nodes. It is well recognized that determining the expansion of a given bipartite graph is a computationally hard problem. Moreover, precise numbers are known only for certain families of graphs. High probability results for expansion are known; however, such results are asymptotic in nature and do not provide deterministic constructions. Parameters such the file size can only be found by inspection of the randomly constructed graph. For these reasons, it is very hard to fully characterize the range of achievable parameters for FR codes (other than the necessary constraints presented above).

Reference [26] presents results on the file size of resolvable FR codes that we have considered above. However, we emphasize that Theorem 19 in [26] does not apply in our situation. For instance, consider the construction of
| Method                     | \((n, \theta, \alpha, \rho)\) | \(\beta\) | \(M\)                                      | Range of \(k\) | Comments                                                                 |
|----------------------------|-------------------------------|------------|--------------------------------------------|----------------|--------------------------------------------------------------------------|
| Affine Resolvable Designs  | \(qp, q^m, q^{n-1}, 1 \leq \rho \leq \frac{q^{n-1}}{q-1}\) | \(q^{m-2}\) | \(q^m\left(1 - \frac{1}{q}\right)^k\) | \(1 \leq k \leq \rho\) | The file size exceeds the trivial lower bound \(k\alpha - \beta_{(2)}\). |
| Hadamard Designs           | \(8a - 2, 4a, 2a, 4a - 1\)    | \(a\)     | \(3a\)                                     | \(1 \leq k \leq 2\) | \(\beta\) can be a composite number.                                    |

**TABLE IX**

Constructions where \(\beta > 1\). In many cases these construction parameters cannot be obtained by trivial \(\beta\)-expansion.

FR codes from MOLS presented above and Lemma 9. Suppose that we choose \(r = p^m - 1\) and \(k = p^m - 2\). In this case it can be verified that for large \(p\), the result of Theorem 19 in [26] does not apply. Furthermore, our affine resolvable design based construction has \(\beta > 1\) and the results of [26] do not apply here.

On a different note, it can also be argued that one can simply treat the FR codes discussed in this section as local codes, by choosing a value of \(k\) that is strictly larger than \(d\) (note that \(k\) is under our control as a system designer). However, we will now argue that this will result in significantly suboptimal codes with respect to the minimum distance bound in Lemma 2. Suppose for instance that we consider a net FR code with parameters \((n = ar, \theta = a^2b, \alpha = ab, \rho = r)\) with \(\beta = b\) and \(d = a\). Note that there are \(r\) parallel classes in the code.

Note that bound in (2), reduces to the Singleton bound as \(d\alpha = a^2b = \theta\), so that \(\lceil \frac{M\alpha}{d\alpha} \rceil = \lceil \frac{M\alpha}{a^2b} \rceil = 1\). Thus, while increasing the value of \(k\) above \(d\) makes the code local, it will be far from the achieving the local code minimum distance bound in Lemma 2. As a concrete example, consider a grid code (an instantiation of the net FR code) with \((n = 20, \theta = 100, \alpha = 10, \rho = 2)\). In this case it can be seen that if \(k = 6\), the code is optimal with respect to the Singleton bound as \(d_{min} = 20 - \lceil \frac{51}{10} \rceil + 1 = 15\). However if choose \(k = 11\), so that it becomes a local code, the corresponding file size is \(M = 80\), so that the minimum distance bound is \(20 - \lceil \frac{80}{10} \rceil + 1 = 13\). However, it can be seen that the code can only recover from at most 9 node failures and not 12. Thus, such a code is a suboptimal local regenerating code. As all the resolvable codes presented in this section are instances of net FR codes, similar statements apply to all these constructions.

Next, consider a FR code \(C\) obtained from a Steiner system with parameters \(\theta, \rho, \alpha, n\) that has a maximal arc of size \(\rho + 1\). In the discussion above we considered the transposed code denoted \(C^T\) and obtained the code rate in Lemma 2 for the range \(1 \leq k \leq \rho + 1\). Note that \(C^T\) has a storage capacity equal to \(\rho\) and is such that any two storage nodes have a unique symbol in common. This implies that the total number of symbols covered by a set of \(\rho + i\) nodes (for \(i > 1\)) is no more than the number of symbols covered by a set of \(\rho + 1\) nodes. Thus, there is no advantage in terms of code rate when we increase \(k\) to more than \(\rho + 1\), or equivalently the code is suboptimal with the local minimum distance bound if \(k\) is increased beyond \(\rho + 1\). In fact, it can be observed that the code is optimal with respect to the Singleton bound only if \(k\) is chosen to be strictly less than \(\rho\).
A failed node can be recovered by contacting 2 nodes and downloading 2 packets from each. The code is resilient up to a total of 3 failures (corresponding to its minimum distance) and the file size is 6.

IV. SOME CHARACTERISTICS OF FR CODES OBTAINED FROM KRONECKER PRODUCTS

It turns out that we can obtain newer constructions of FR codes by appropriately combining existing constructions using the Kronecker product. In this section we discuss the properties of the newly obtained codes. Under certain conditions we can compute the file size of the new codes, though in general it turns out to be a hard problem. We also show that combining certain FR codes yields FR codes with highest possible resilience to node failures and that combining resolvable FR codes yields resolvable FR codes.

We begin with a simple example that generates a code that meets the Singleton bound. Let \( \theta = 2a + 1 \) for \( a \geq 1 \) and the incidence matrices \( N_1 \) and \( N_2 \) be equal to \( J - I \) where \( J \) denotes \( \theta \times \theta \) all-ones matrix and \( I \) denotes the identity matrix of the appropriate size. Then, the FR code \( C \) obtained from the incidence matrix \( \bar{N} = N_1 \otimes N_2 \) has the following properties:

- The parameters of the code are \( \bar{n} = \theta = (2a + 1)^2 \) and \( \bar{a} = \rho = (2a)^2 \).
- A failed node can be recovered by contacting 2 nodes.
- Contacting any two nodes recovers at least \( 2a(2a + 1) \) symbols. Thus, when \( k = 2 \), we have that the file size \( \mathcal{M} = 2a(2a + 1) \), where it can be observed that \( \lceil \frac{\mathcal{M}}{\alpha} \rceil = 2 \), so that the code meets the Singleton bound.

**Example 11:** Let \( C = (\Omega, V) \) be a FR code with \( \Omega = \{1, 2, 3\} \) and \( V = \{V_1 = \{2, 3\}, V_2 = \{1, 3\}, V_3 = \{1, 2\}\} \), so that its incidence matrix \( N = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix} \). The new code is obtained from the incidence matrix of \( \bar{N} = N \otimes N \) and the storage nodes are shown in Fig. 8.

Suppose that the outer MDS code has parameters \( (9, 6) \), so that \( \theta = 9, \mathcal{M} = 6 \). It can be observed in this construction that the file can be recovered by contacting any two nodes, so that \( k = 2 \) and that a failed node can be recovered by contacting two nodes and downloading two packets from each of them.

**Lemma 10:** Let \( C_1 = (\Omega_1, V_1) \) and \( C_2 = (\Omega_2, V_2) \) be two FR codes with parameters \( (n_1, \theta_1, \alpha, \rho_1) \) and \( (n_2, \theta_2, \alpha, \rho_2) \) such that any two storage nodes in \( C_1 \) (or \( C_2 \)) have at most one symbol in common. Let \( \mathcal{M}_1 \) and \( \mathcal{M}_2 \) denote the file sizes of \( C_1 \) and \( C_2 \) respectively for a given \( k_1 \leq \min \{n_1, n_2\} \). Suppose either \( \mathcal{M}_1 \) or \( \mathcal{M}_2 \) is equal to \( k_1\alpha - \binom{k_1}{2} \). Then the FR code \( C \) obtained from Kronecker product of \( C_1 \) and \( C_2 \) has parameters \( (n = n_1n_2, \theta = \theta_1\theta_2, \alpha^2, \rho_1\rho_2) \). The file size for \( C \) when \( k = k_1 \) is given by \( k_1\alpha^2 - \alpha \binom{k_1}{2} \).

**Proof:** Let \( N_1 \) and \( N_2 \) denote the incidence matrices of the FR codes \( C_1 \) and \( C_2 \). Let \( c_i \) denote a column in \( N_1 \) and \( d_i \) denote a column in \( N_2 \). The overlap between any two columns in \( N_1 \otimes N_2 \) can be expressed as \( (c_i \otimes d_j)^\dagger(c_i \otimes d_j') = c_i^\dagger c_i' \otimes d_j^\dagger d_j' \leq \alpha \). Thus the overlap between any two columns in \( N_1 \otimes N_2 \) is at most \( \alpha \) and therefore the file size of \( C \) is at least \( k_1\alpha^2 - \alpha \binom{k_1}{2} \).
We know that any two nodes in \( C_1 \) and \( C_2 \) have at most one symbol in common. Thus, using a simple inclusion-exclusion principle argument it can be seen that \( M_i \geq k_1 \alpha - \binom{k_i}{2} \) for \( i = 1, 2 \). Furthermore we are given that one of them meets this lower bound. Without loss of generality we assume that \( M_1 = k_1 \alpha - \binom{k_i}{2} \). This implies that there exists a set of column vectors \( I_1 = \{c_1, \ldots, c_{k_1}\} \) in \( N_1 \) such that they cover \( M_1 = k_1 \alpha - \binom{k_i}{2} \) symbols, i.e., any two columns from \( I_1 \) have exactly one symbol in common and any three columns from \( I_1 \) have no symbols in common (see Appendix).

Next, we demonstrate a set of columns in \( N_1 \otimes N_2 \) that meets this lower bound. Let us consider a column in \( N_2 \), denoted \( d_1 \) and examine \( N_1 \otimes d_1 \). Within, this set we have a subset of \( k_1 \) columns denoted \( I_2 = \{c_i \otimes d_1, \text{for } c_i \in I_1\} \). Now \((c_i \otimes d_1)(c_j \otimes d_1) = c_i c_j \otimes d_1^T d_1 = \alpha\), whereas any three columns from \( I_2 \) will have a zero overlap. Thus, the number of symbols covered by this set is exactly \( k_1 \alpha^2 - \alpha \binom{k_i}{2} \).

This Lemma can be used to determine the file size for the Kronecker product of certain Steiner systems.

**Lemma 11:** Let \( C \) be a FR code obtained from a Steiner system \( S(2, \alpha, \theta) \) with \( \rho = \frac{\theta - 1}{\alpha - 1} \) such that it has a maximal arc of size \( \rho + 1 \). Then the Kronecker product of the transposed code \( C^T \) with itself is such that the file size equals \( kp^2 - \rho \binom{k_i}{2} \) for \( 1 \leq k \leq \rho \).

**Proof:** The result follows from Lemma 3 and Lemma 10.

**Remark 5:** By Skolem’s construction [35] we have \( S(2, 3, \theta) \) for all \( \theta \geq 7 \) and \( \theta \equiv 1, 3 \mod 6 \). Moreover, for all \( \theta \geq 7 \) and \( \theta \equiv 3, 7 \mod 12 \) a Steiner system \( S(2, 3, \theta) \) has at least one maximal arc [36]. Thus, Lemma 11 applies.

**Lemma 12:** Let \( N_1 \) and \( N_2 \) be incidence matrices of two FR codes such that the size of the pairwise intersection of distinct nodes is at most 1. Let \((n_1, \theta_1, \alpha, \rho_1)\) and \((n_2, \theta_2, \alpha, \rho_2)\) be parameters of these FR codes respectively. Assume that the FR code obtained from \( \bar{N} = N_1 \otimes N_2 \) has normalized repair bandwidth \( \beta = \alpha \). Then the FR code \( \bar{N} \) is resilient up to \( \rho_1 \rho_2 - 1 \) failures.

**Proof:** Define \( \mathcal{N}(c_i) \) (\( \mathcal{N}(d_j) \)) to be the set of storage nodes in \( N_1 \) (\( N_2 \)) that have exactly one symbol in common with \( c_i \) (\( d_j \)). As \( N_1 \) and \( N_2 \) are Steiner systems, two nodes have at most one symbol in common.

In the discussion below we show that if there are at most \( \rho_1 \rho_2 - 1 \) failures, we can recover all the nodes. We proceed by contradiction, i.e., assume that there exists a set of failed nodes \( F^* \) in \( \bar{N} \) with \( |F^*| = \rho_1 \rho_2 - 1 \). Suppose that there is a failed node \( c_i \otimes d_j \in F^* \) that cannot be recovered. Note that \( \beta = \alpha \). Thus, we need to download \( \alpha \) symbols each from the surviving nodes, i.e., we need to consider nodes in \( \bar{N} \) that have an overlap of \( \alpha \) with \( c_i \otimes d_j \).

Our first observation is that only the nodes in \( \mathcal{N}(c_i) \otimes d_j \) and \( c_i \otimes \mathcal{N}(d_j) \) are useful for recovering \( c_i \otimes d_j \). To see this consider a node \( c_i' \otimes d_j' \) in \( \bar{N} \) such that it does not belong to \( \mathcal{N}(c_i) \otimes d_j \) or \( c_i \otimes \mathcal{N}(d_j) \). If \( c_i' = c_i \), then \( d_j' \notin \mathcal{N}(d_j) \), i.e., \( (c_i' \otimes d_j')^T (c_i \otimes d_j) = 0 \); a similar argument holds when \( c_i' \notin \mathcal{N}(c_i) \), \( d_j' = d_j \). Otherwise \( (c_i' \otimes d_j')^T (c_i \otimes d_j) \) can be at most 1. Thus, only the nodes in \( \mathcal{N}(c_i) \otimes d_j \) and \( c_i \otimes \mathcal{N}(d_j) \) are useful for reconstructing \( c_i \otimes d_j \).

Next, note that \( c_i \) (\( d_j \)) can be expressed as the sum of \( \alpha \) unit vectors of length \( \theta_1 \) (\( \theta_2 \)). Let \( e_k \) denote the unit vector with a one in the \( k \)-th location. Thus, \( c_i = \sum_{k \in I_1} e_k \), where \( I_1 \subset [\theta_1] \) and \( d_j = \sum_{l \in I_2} e_l \) where \( I_2 \subset [\theta_2] \). Thus, the overlap between \( c_i \otimes d_j \) and \( \mathcal{N}(c_i) \otimes d_j \) can be expressed as \( e_k \otimes d_j \) for some \( k \in I_1 \). A similar statement holds for the overlap between \( c_i \otimes d_j \) and \( c_i \otimes \mathcal{N}(d_j) \). Our next observation is that when we reconstruct \( c_i \otimes d_j \), we can either download symbols from \( \mathcal{N}(c_i) \otimes d_j \) or from \( c_i \otimes \mathcal{N}(d_j) \) but not both. Indeed, for \( k \in I_1, l \in I_2 \), we have \( (c_i \otimes e_l)^T (e_k \otimes d_j) = 1 \). Thus, if we download symbols from both \( \mathcal{N}(c_i) \otimes d_j \) and from \( c_i \otimes \mathcal{N}(d_j) \), then we
will need to download strictly more than $\alpha^2$ symbols for reconstructing $c_i \otimes d_j$.

Note that there are $p_1$ copies of each $e_k \otimes d_j$, where $k \in I_1$. If there is at least one copy of $e_k \otimes d_j$, for all $k \in I_1$ available in the surviving nodes, then it is clear that $c_i \otimes d_j$ can be recovered by downloading copies of each $e_k \otimes d_j$ from the surviving nodes. Likewise, there are $\rho_2$ copies of each $c_i \otimes e_l$ for $l \in I_2$ and $c_i \otimes d_j$ can be recovered if each of these copies is available in the surviving nodes. In the discussion below we say that $c_i \otimes d_j$ is recoverable if either or both of these situations apply.

Thus, it is clear that if $c_i \otimes d_j$ is not recoverable it has to be the case that all copies of $e_k \otimes d_j$ for some $k^* \in I_1$ are unavailable. This implies that there exists a set of failed nodes denoted $F_1 \subset \mathcal{N}(c_i) \otimes d_j$ of size at least $\rho_1 - 1$.

Arguing in a similar vein, we can consider whether $c_i \otimes d_j$ can be recovered from the nodes in $c_i \otimes \mathcal{N}(d_j)$. Based on the discussion above, if $c_i \otimes d_j$ is not recoverable, it has to be the case that there exists a set of failed nodes $F_2 \subset c_i \otimes \mathcal{N}(d_j)$ of size at least $\rho_2 - 1$. In addition the node sets $\mathcal{N}(c_i) \otimes d_j$ and $c_i \otimes \mathcal{N}(d_j)$ are disjoint, thus $F_1 \cap F_2 = \emptyset$, i.e., it is clear that there are at least $\rho_1 + \rho_2 - 2$ failures are essential to ensure that $c_i \otimes d_j$ is not recoverable.

Next, we examine whether any of the nodes in $F_1 \cup F_2$ are recoverable. A given node in $F_1$ is of the form $c_i \otimes d_j$ where $c_i \otimes d_j = 1$. It is evident that $c_i \otimes d_j$ cannot be recovered from $\mathcal{N}(c_i) \otimes d_j$ as all copies of $e_k \otimes d_j$ for a specific $k^*$ are unavailable owing to the failure of the nodes in $F_1$. Specifically, note that it rules out the possibility of using the surviving nodes in the set $\mathcal{N}(c_i) \otimes d_j$. From the previous observation, it can only be recovered exclusively from the nodes in $c_i \otimes \mathcal{N}(d_j)$.

Thus, there need to be at least $\rho_2 - 1$ failures from the node set $c_i \otimes \mathcal{N}(d_j)$ to ensure that $c_i \otimes d_j$ is not recoverable. Furthermore, these failures are distinct from the failures in $F_1 \cup F_2$. Arguing in this way for each node in $F_1$, we conclude that at least $(\rho_1 - 1)(\rho_2 - 1)$ failures need to be induced to ensure that none of the nodes in $F_1$ can be recovered.

However, this implies a total of $1 + \rho_1 + \rho_2 - 2 + (\rho_1 - 1)(\rho_2 - 1) = \rho_1 \rho_2 > \rho_1 \rho_2 - 1$ failures. Thus, we conclude that even if an appropriate $F_1 \cup F_2$ can be found for $c_i \otimes d_j$, at least one node in $F_1$ can be recovered. After this recovery, the set $F_1$ cannot exist. This implies that $c_i \otimes d_j$ can be recovered. As the choice of $c_i \otimes d_j$ was arbitrary, we can recover any node when there are at most $\rho_1 \rho_2 - 1$ failures.

This bound is tight since each symbol in $\tilde{N}$ is repeated $\rho_1 \rho_2$ times. Thus, we can easily find a set of $\rho_1 \rho_2$ failures that we cannot recover from.

\textbf{Corollary 2:} Let $N_1$ and $N_2$ be transposes of incidence matrices of two Steiner systems namely $S(2, \alpha_1, \theta_1)$ and $S(2, \alpha_2, \theta_2)$ where the parameters satisfy $\rho = \frac{\theta_1 - 1}{\alpha_1 - 1} = \frac{\theta_2 - 1}{\alpha_2 - 1}$. Assume the FR code obtained from $\tilde{N}$ has normalized repair bandwidth $\beta = \rho$. Then, the FR code is resilient up to $\alpha_1 \alpha_2 - 1$ failures.

\textbf{Proof:} Any two nodes meet in exactly one symbol in the FR code obtained by transposes of incidence matrices of a Steiner system. Also note that the main ingredient of the proof Lemma 12 is the property that two nodes meet in at most one symbol in Steiner systems. So the rest follows similarly as in the previous proof.

We also investigate the properties of FR codes that are generated by taking the Kronecker product of net FR codes with themselves. The Kronecker product does not necessarily produce a new net FR codes but it yields a resolvable FR code. For example, in Fig. 9 a resolvable FR code obtained from Kronecker product of a net FR code with itself.

\textbf{Lemma 13:} Let $N$ be the incidence matrix of a net FR code with parameters $(n, \theta, \alpha, \rho)$. Then, the FR code
Fig. 9. The resultant FR code has \( \Omega = \{1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F, G\} \). Each storage node contains 4 symbols. A failed node can be recovered by contacting two nodes and downloading 2 packets from each. The code is resilient up to 3 failures.

obtained from \( \bar{N} = N \otimes N \) is a resolvable FR code.

**Proof:** We can order the columns of \( N \) with respect to the \( \rho \) parallel classes. Assume that the \( j \)-th block in \( i \)-th parallel class is represented by the column \( c_{i,j} \). We will show for fixed \( i \) and \( s \), \( c_{i,j} \otimes c_{s,r} \) with \( 1 \leq j \leq \frac{\theta}{\alpha} \) and \( 1 \leq r \leq \frac{\theta}{\alpha} \) forms a set of blocks which is a parallel class. There will be \( \theta^2 \alpha^2 \) blocks in this set, hence it is enough to show any distinct two blocks does not share any points. Since \( (c_{i,j} \otimes c_{s,r})(c_{i,u} \otimes c_{s,v}) = (c_{i,j}c_{i,u} \otimes c_{s,r}c_{s,v}) \) equals the zero, the \( \frac{\theta^2 \alpha^2}{\alpha^2} \) vectors form a parallel class.

**Example 12:** A simple example can be obtained from \( C = (\Omega, V) \) where \( \Omega = \{1, 2, 3, 4\} \) and \( V = \{V_1 = \{1, 2\}, V_2 = \{3, 4\}, V_3 = \{1, 3\}, V_4 = \{2, 4\}\} \). The code obtained from \( \bar{N} = N \otimes N \) is illustrated in Fig. 9.

V. Construction of FR Codes when \( d < k \)

In the discussion so far, we have considered FR codes where the recovery degree \( d \geq k \), i.e., the repair degree \( (d) \) of the code is at least as high as the number of nodes \( (k) \) contacted for recovering the file. Of course, the codes operate at the MBR point which implies that they download exactly \( \alpha \) symbols for regeneration. However, as discussed in Section I, in many application scenarios it has been recognized that the number of nodes that the new node has to contact is an important metric that needs to be optimized, as against the repair bandwidth. Note that the definition of a FR code does not rule out codes where \( d < k \).

In this section, we discuss constructions of **locally recoverable FR codes** that have the property that \( d < k \). It turns out that the minimum distance bound for locally recoverable codes that was derived in [3], [4], needs to be refined for our scenario of exact, uncoded and table-based repair. We derive such a bound and present constructions that meet this bound. 

**Definition 12:** **Locally recoverable fractional repetition code.** Let \( C = (\Omega, V) \) be a FR code for a \((n, k, d, \alpha)\)-DSS, with repetition degree \( \rho \) and normalized repair bandwidth \( \beta = \alpha/d \). If the repair degree \( d < k \), then the FR code \( C \) is called a locally recoverable fractional repetition code.

As before we define \( \rho_{res} \) to be the maximum number of node failures such that each failed node can be recovered by contacting \( d \) surviving nodes and downloading symbols from them. For a node \( V_i \in V \) in \( C \), let \( S(V_i) \in V \) denote the set of nodes ( with \( |S(V_i)| < k \) ) that are contacted if \( V_i \) fails. We refer to \( S(V_i) \) as the local structure associated with \( V_i \). Note that it is possible that the set of nodes in \( S(V_i) \) and the corresponding symbols form a FR code (cf. Definition 2); however this is not essential.
A. Codes for systems with $\rho_{res} = 1$

Our first construction is a class of codes which is optimal with respect to the bound provided in Lemma \ref{lem:minimum-distance-bound} and allow local recovery in the presence of a single failure. Our construction leverages the properties of graphs with large girth.

**Definition 13:** An undirected graph $\Gamma$ is called an $(s, g)$-graph if each vertex has degree $s$, and the length of the shortest cycle in $\Gamma$ is $g$.

**Construction 1:** Let $\Gamma = (V', E')$ be a $(s, g)$-graph with $|V'| = n$.

(i) Arbitrarily index the edges of $\Gamma$ from 1 to $\frac{n^2}{2}$.

(ii) Each vertex of $\Gamma$ corresponds to a storage node and stores the symbols incident on it.

It can be observed that the above procedure yields an FR code $C = (\Omega, V)$ with $n$ storage nodes, parameters $\theta = \frac{n^2}{2}$, $\alpha = s$ and $\rho = 2$. Upon single failure, the failed node can be regenerated by downloading one symbol each from the storage nodes corresponding to the vertices adjacent to it in $\Gamma$ (i.e., $\beta = 1$); thus, the repair degree $d = s$. Note that for this construction, the local structures are typically not FR codes. Suppose that the storage node corresponding to vertex $v_1 \in \Gamma$ fails, then we contact the storage nodes corresponding to its $(s - 1)$ neighbors in $\Gamma$; this is the local structure associated with $v_1$. If the girth $g > 3$, then it is clear that the nodes in the local structure do not have symbols in common, i.e., they do not form a FR code.

**Remark 6:** We note that the work of \cite{7} also used the above construction for MBR codes; however, they did not have the girth restriction on $\Gamma$. As we discuss next, $(s, g)$-graphs allow us to construct locally recoverable codes and provide a better bound on the file size when $k \leq g$. We allow the system parameter $k$ to be greater than $d$, however in the work of \cite{7}, they consider only the case $k \leq d$. The work of \cite{26} also used high-girth graphs, but their constructions are not in the context of locally recoverable codes.

**Lemma 14:** Let $C = (\Omega, V)$ be a FR code constructed by Construction 1. If $s > 2$, and $k \leq g$, we have $|\bigcup_{i=1}^{k} V_i| \geq k(s - 1)$ for any $V_i \in V, i = 1, \ldots k$.

**Proof:** Let $V_1, V_2, \ldots, V_{k-1}$ and $V_k$ be any $k$ nodes in our DSS, where $k \leq g$. We argue inductively. Note that $|V_1| = s > s - 1$. Suppose that $|\bigcup_{i=1}^{j} V_i| \geq j(s - 1) + \xi$ for $j < k$, where $0 \leq \xi \leq j$ is the number of connected components formed by the nodes $V_1, \ldots, V_j$ in $\Gamma$. Now consider $|\bigcup_{i=1}^{j+1} V_i|$ where $j + 1 < k$. Note that since $j + 1 < g$ there can be no cycle in $\bigcup_{i=1}^{j+1} V_i$. Thus, $V_{j+1}$ is connected to at most once to each connected component in $\bigcup_{i=1}^{j} V_i$. Suppose that $V_{j+1}$ is connected to $\ell$ existing connected components in $\bigcup_{i=1}^{j} V_i$, where $0 \leq \ell \leq \min(\xi, s)$. Then, the number of connected components in $\bigcup_{i=1}^{j+1} V_i$ is $\xi - \ell + 1$ and the number of new symbols that it introduces is $s - \ell$. Therefore $|\bigcup_{i=1}^{j+1} V_i| = j(s - 1) + \xi + s - \ell = (j + 1)(s - 1) + \xi - \ell + 1$. This proves the induction step.

Thus, $|\bigcup_{i=1}^{k-1} V_i| \geq (k - 1)(s - 1) + \xi_{k-1}$, where $\xi_{k-1}$ is the number of connected components formed by $V_1, \ldots, V_{k-1}$. Now consider $\bigcup_{i=1}^{k} V_i$. Note that there can be a cycle introduced at this step if $k = g$. Now, if $\xi_{k-1} \geq 2$, it can be seen that $V_k$ can only connect to each of the $\xi_{k-1}$ connected components once, otherwise it would imply the existence of a cycle of length strictly less than $g$ in $\Gamma$. Thus, in this case $|\bigcup_{i=1}^{k} V_i| \geq k(s - 1)$. On the other hand if $\xi_{k-1} = 1$, then $V_k$ can connect at most twice to this connected component. In this case again we can observe that $|\bigcup_{i=1}^{k} V_i| \geq k(s - 1)$.

**Lemma 15:** Let $\Gamma = (V, E)$ be a $(s, g)$-graph with $|V| = n$ and $s > 2$. If $g \geq k = as + b$ such that $s > b \geq a + 1$, then $C$ obtained from $\Gamma$ by Construction 1 is optimal with respect to the minimum distance bound in Lemma \ref{lem:minimum-distance-bound} when the file size $M = k(s - 1)$. 
Proof: From Lemma 14, any \( k \) nodes cover at least \( k(s-1) \) symbols. Thus, the code is minimum distance optimal when the statement in Observation 1 holds, i.e.,

\[
\left\lceil \frac{k(s-1)}{s} \right\rceil + \left\lceil \frac{k(s-1)}{s^2} \right\rceil = k + 1.
\]

We have

\[
k(s-1) = (as + b)(s-1) = as^2 + (b-a)s - b.
\]

Since, \( s > b \geq a + 1 \) the following holds.

\[
\left\lceil \frac{k(s-1)}{s} \right\rceil = \left\lfloor \frac{as^2 + (b-a)s - b}{s} \right\rfloor = as + (b-a),
\]
and

\[
\left\lfloor \frac{as^2 + (b-a)s - b}{s^2} \right\rfloor = \left\lfloor a + \frac{(b-a)s - b}{s^2} \right\rfloor \geq a + 1.
\]

Corollary 3: Let \( \Gamma = (V, E) \) be a \((s, g)\)-graph with \( |V| = n \) and \( s > 2 \). If \( g \geq s + 2 \), then \( C \) obtained from \( \Gamma \) by Construction 1 is optimal with respect to the bound in Lemma 2 for file size \( M = s^2 + s - 2 \).

It can be observed that in the specific case of \( s = 2 \), applying Construction 1 results in a DSS where the union of any \( k \) nodes has at least \( k + 1 \) symbols. We now discuss some examples of codes that can be obtained from our constructions.

Sachs [37] provided a construction which shows that for all \( s, g \geq 3 \), there exists a \( s \)-regular graph of girth \( g \). Also, explicit constructions of graphs with arbitrarily large girth are known [38]. Using these we can construct infinite families of optimal locally recoverable codes.

Example 13: The Petersen graph on 10 vertices and 15 edges can be shown to be a \((3, 5)\)-graph. We label the edges 1, \ldots, 10 and \( A, B, \ldots, E \) in Fig. 10. Let the file size \( M = 3^2 + 3 - 2 = 10 \); we use a \((15, 10)\) outer MDS code. Applying Construction 1, we obtain a DSS with parameters \( n = 10, k = 5, \alpha = 3, \rho = 2, r = 3 \). From Corollary 3 we observe that the DSS meets the minimum distance bound.

An \((s, g)\)-graph with the fewest possible number of vertices, among all \((s, g)\)-graphs is called an \((s, g)\)-cage and will result in the maximum code rate for our construction. For instance, the \((3, 5)\)-cage is the Petersen graph.

We note here that bipartite cages of girth 6 were used to construct FR codes in [24] though these were not in the context of locally recoverable codes.

B. Codes for systems with \( \rho_{res} > 1 \)

Our second class of codes are such that the local structures are also FR codes. The primary motivation for considering this class of codes is that they naturally allow for local recovery in the presence of more than one failure as long as the local FR code has a repetition degree greater than two. Thus, in these codes, each storage node participates in one or more local FR codes that allow local recovery in the presence of failures. For these classes of codes, we can derive the following tighter upper bound on the minimum distance when the file size is larger than the number of symbols in one local structure.

Lemma 16: Let \( C \) be a locally recoverable FR code with parameters \((n, \theta, \alpha, \rho)\) where each node belongs to a local FR code with parameters \((n_{loc}, \theta_{loc}, \alpha, \rho_{loc})\). Suppose that the file size \( M > \theta_{loc} \). Then,

\[
d_{min} \leq \max \left( n - \left\lceil \frac{M\rho_{loc}}{\alpha} \right\rceil + \rho_{loc}, \ n + n_{loc} + 1 - \left\lceil \frac{M\rho_{loc} + \theta_{loc}}{\alpha} \right\rceil \right).
\]
Fig. 10. The figure shows the Petersen graph with its edges labeled from 1, \ldots, 10 and A, \ldots, E. Each vertex acts as a storage node and stores the symbols incident on it.

**Proof:**

We will apply an algorithmic approach here (inspired by the one used in [3]). Namely, we iteratively construct a large enough set $S \subset V$ so that $|S| < M$. The minimum distance bound is then given by $n - |S|$. Our algorithm is presented in Fig. [1][1] Towards this end, let $S_i$ and $H(S_i)$ represent the number of nodes and the number of symbols included at the end of the $i$-th iteration. Furthermore, let $s_i = |S_i| - |S_{i-1}|$ and $h_i = |H(S_i)| - |H(S_{i-1})|$, represent the corresponding increments between the $(i - 1)$-th and the $i$-th iteration. We divide the analysis into two cases.

- **Case 1:** [The algorithm exits without ever entering line 8.] Note that we have $1 \leq s_i \leq n_{loc}$ and $h_i \leq \theta_{loc} - a(n_{loc} - s_i)$ where $a(n_{loc} - s_i)$ is the minimum number of symbols covered by $(n_{loc} - s_i)$ nodes in the local FR code and hence a lower bound on $|\Omega_{Fr} \cap H(S_{i-1})|$. By considering the bipartite graph representing the local FR code it can be seen that $a(n_{loc} - s_i) \geq \frac{(n_{loc} - s_i)\alpha}{\rho_{loc}}$. Thus, we have

$$\theta_{loc} - a(n_{loc} - s_i) \leq \theta_{loc} - \frac{n_{loc}\alpha - s_i\alpha}{\rho_{loc}} = \frac{s_i\alpha}{\rho_{loc}}.$$  

Suppose that the algorithm runs for $l$ iterations and exits on the $l + 1$ iteration. Then

$$\sum_{i=1}^{l} s_i \geq \frac{\rho_{loc}}{\alpha} \sum_{i=1}^{l} h_i.$$  

Since the algorithm exits without ever entering line 8, it is unable to accumulate even one additional node. Hence

$$\sum_{i=1}^{l} h_i \geq \mathcal{M} - \alpha,$$  

which implies

$$\sum_{i=1}^{l} s_i \geq \left\lceil \frac{\rho_{loc}}{\alpha} (\mathcal{M} - \alpha) \right\rceil$$  

by the integer constraint.
1: \(S_0 = \emptyset, \ i = 1\)

2: \textbf{while} \(H(S_{i-1}) < \mathcal{M}\) \textbf{do}

3: \quad \text{For each node} \(Y_j \in S_{i-1}\), identify an FR code \(Pf_j = (\Omega_{Pf_j}, V_{Pf_j})\) (if it exists) such that \(Y_j \in V_{Pf_j}, V_{Pf_j} \nsubseteq S_{i-1}\). If no such FR code exists, find an FR code that has no intersection with \(S_{i-1}\) and set \(Pf_1\) equal to it.

   \quad \bullet \text{Let} \(b_j = \left|\Omega_{Pf_j} \cap H(S_{i-1})\right|\). Let \(j^* = \arg \max \ b_j\).

4: \quad \text{if} \(\theta_{loc} - b_{j^*} + H(S_{i-1}) \geq \mathcal{M}\) \textbf{then}

5: \quad \quad \text{Set} \(S_i = S_{i-1} \cup V_{Pf_{j^*}}\).

6: \quad \textbf{else}

7: \quad \quad \text{if there exists} \ A \subset V_{Pf_{j^*}} \text{ such that} \ |S_{i-1} \cup A| > |S_{i-1}| \text{ and } H(S_{i-1} \cup A) \leq \mathcal{M}\) \textbf{then}

8: \quad \quad \quad \text{Let} \(V_{Pf'_{j^*}} = \arg \max_{A \subset V_{Pf_{j^*}}} H(S_{i-1} \cup A) < \mathcal{M}\). Set \(S_i = S_{i-1} \cup V_{Pf'_{j^*}}\).

9: \quad \quad \text{else}

10: \quad \quad \quad \text{Exit.}

11: \quad \textbf{end if}

12: \textbf{end if}

13: \textbf{end while}

Fig. 11. Algorithm for finding the distance bound

Thus, the bound on the minimum distance becomes

\[
d_{min} \leq n - \left\lceil \frac{\rho_{loc} \mathcal{M}}{\alpha} \right\rceil + \rho_{loc}.
\]

\textbf{Case 2:} [The algorithm exits after entering line 8.] Note that by assumption, \(\mathcal{M} > \theta_{loc}\). Suppose that the algorithm enters line 5, \(l \geq 1\) times. Now we have \(\sum_{i=1}^{l} h_i \geq \mathcal{M} - \theta_{loc}\), otherwise we could include another local structure. Hence we need to add nodes so that strictly less than \(\mathcal{M} - \sum_{i=1}^{l} h_i\) symbols are covered. It can be seen that we can include at least

\[
\left\lceil \frac{\mathcal{M} - \sum_{i=1}^{l} h_i}{\alpha} \right\rceil - 1
\]

more nodes. Therefore, the total number of nodes accumulated is

\[
\geq \frac{\rho_{loc}}{\alpha} \sum_{i=1}^{l} h_i + \left\lceil \frac{\mathcal{M} - \sum_{i=1}^{l} h_i}{\alpha} \right\rceil - 1
\]

\[
\geq \frac{\rho_{loc} - 1}{\alpha} (\mathcal{M} - \theta_{loc}) + \frac{\mathcal{M}}{\alpha} - 1
\]

\[
= \frac{\mathcal{M} \rho_{loc} + \theta_{loc}}{\alpha} - n_{loc} - 1.
\]

Therefore, we have the following minimum distance bound

\[
d_{min} \leq n + n_{loc} + 1 - \left\lceil \frac{\mathcal{M} \rho_{loc} + \theta_{loc}}{\alpha} \right\rceil.
\]

The final bound is obtained by taking the maximum of the two bounds obtained above.

\[\square\]

The following corollary can be also be established.
Corollary 4: Let $C$ be a locally recoverable FR code with parameters $(n, \theta, \alpha, \rho)$ where each node belongs to a local FR code with parameters $(n_{loc}, \theta_{loc}, \alpha_{loc}, \rho_{loc})$. Furthermore, suppose that $C$ can be partitioned as the union of $\ell$ disjoint local FR codes. If the file size $M = t\theta_{loc} + \beta$ for some integer $1 \leq t < \ell$ and $\beta \leq \alpha$, we have $d_{min} \leq n - \left\lceil \frac{M\rho_{loc}}{\alpha_{loc}} \right\rceil + \rho_{loc}$.

Proof: Applying the algorithm in Fig. 11 it can be observed that we will never enter line 8, as $C$ consists of the union of disjoint local FR codes and the file size $M = t\theta_{loc} + \beta$. Thus, after accumulating $t$ disjoint local FR codes, the algorithm will exit, yielding the required bound. □

Construction 2: Let $C = (\Omega, V)$ be a FR code with parameters $(n, \theta, \alpha, \rho)$ such that any $\Delta + 1$ nodes in $V$ cover $\theta$ symbols and for $V_i, V_j \in V$, we have $|V_i \cap V_j| \leq \beta$ when $i \neq j$. We construct a locally recoverable FR code $\bar{C}$ by considering the disjoint union of $l(>1)$ copies of $C$. Thus, $\bar{C}$ has parameters $(ln, l\theta, \alpha, \beta)$. We call $\bar{C}$ the local FR code of $\bar{C}$.

Lemma 17: Let $\bar{C}$ be a code constructed by Construction 2 for some $l > 1$ such that the parameters of the local FR code satisfy $(\rho - 1)\alpha \theta - (\theta + \alpha)(\Delta - 1)\beta \geq 0$. Let the file size be $M = t\theta + \alpha$ for some $1 \leq t < l$. Then $\bar{C}$ is optimal with respect to Corollary 4.

Proof: It is evident that $\bar{C}$ is the disjoint union of $l$ local FR codes. Thus, the minimum distance bound here is $d_{min} \leq ln - \left\lceil \frac{l(\theta + \alpha)\rho}{\alpha} \right\rceil + \rho = (l-t)n$. The code is optimal when any $tn + 1$ nodes in $\tilde{C}$ cover at least $M = t\theta + \alpha$ symbols. We show that this is the case below.

Let $a_i$ be the number of nodes that are chosen from the $i$-th local FR code and $X_i$ be the symbols covered by these $a_i$ nodes. Note that for any $1 \leq i \leq l$ if $a_i \geq \Delta + 1$, then $X_i = \theta$ (the maximum possible). Suppose there are $0 \leq t_1 \leq t$ local FR codes that cover $\theta$ symbols. It can be seen that in this case it suffices to show that $(t - t_1)n + 1$ nodes cover at least $(t - t_1)\theta + \alpha$ symbols. Here we can omit case of $t = t_1$, since our claim clearly holds in this situation. Suppose that these nodes belong to $s$ local FR codes, where $a_i \leq \Delta, i = 1, \ldots, s$. By applying Corradi’s lemma [39] we obtain

$$|X_i| \geq \frac{\alpha^2 a_i}{\alpha + (a_i - 1)\beta} \geq \frac{\alpha^2 a_i}{\alpha + (\Delta - 1)\beta}.$$  

This implies that

$$\sum_{i=1}^{s} |X_i| \geq \sum_{i=1}^{s} \frac{\alpha^2 a_i}{\alpha + (\Delta - 1)\beta}$$

$$= \frac{\alpha^2}{\alpha + (\Delta - 1)\beta} \sum_{i=1}^{s} a_i$$

$$= \frac{\alpha^2}{\alpha + (\Delta - 1)\beta} ((t - t_1)n + 1)$$

$$= \frac{(t - t_1)\theta \rho \alpha}{\alpha + (\Delta - 1)\beta} + \frac{\alpha^2}{\alpha + (\Delta - 1)\beta} \quad \text{(since } n\alpha = \theta \rho\text{)}$$

$$= (t - t_1)\theta + \left( \frac{\rho \alpha}{\alpha + (\Delta - 1)\beta} - 1 \right) (t - t_1)\theta + \frac{\alpha^2}{\alpha + (\Delta - 1)\beta}$$

$$\geq (t - t_1)\theta + \frac{(\rho - 1)\alpha - (\Delta - 1)\beta}{\alpha + (\Delta - 1)\beta} \theta + \frac{\alpha^2}{\alpha + (\Delta - 1)\beta}$$

$$\geq (t - t_1)\theta + \alpha \quad \text{(using the assumed conditions)}$$
The above lemma can be used to generate several examples of locally recoverable codes with $\rho_{res} > 1$. We discuss two examples below.

**Example 14:** Let $q$ be a prime power. We consider the codes obtained from affine resolvable designs discussed in Section III-B.1. These codes have parameters $\theta = q^m, \alpha = q^{m-1}, \rho = \frac{q^{m-1}}{q-1}$ and $n = q\rho$. These codes are resolvable and hence we can vary the repetition degree by choosing an appropriate number of parallel classes. Note that the number of nodes in a parallel class is $\theta/\alpha = q$.

Suppose we choose the local FR code by including $q^{m-1}$ parallel classes, so that the repetition degree is $q^{m-1}$ and there are $n = q^m$ nodes. Furthermore, it can be shown that $\beta = q^{m-2}$. The value of $\Delta$ (cf. Definition 2) can be determined as follows. For the local FR code, any subset of at least $q^m - q^{m-1} + 1$ nodes has at least one intact parallel class, which covers all the $\theta = q^m$ symbols. Accordingly, for this code we can conclude that $\Delta = q^m - q^{m-1}$.

Next, we verify the conditions of Lemma 17. For this local FR code, we have that

$$\begin{align*}
(\rho - 1)\alpha\theta - (\theta + \alpha)(\Delta - 1)\beta &= (q^{m-1} - 1)q^{2m-1} - (q^m + q^{m-1})(q^m - q^{m-1} - 1)q^{m-2} \\
&= q^{2m-3}(q^{m+1} - q^2 - (q + 1)(q^m - q^{m-1} - 1)) \\
&= q^{2m-3}(q^{m+1} - q^2 - (q^m - q^{m-1} - 1) - (q^{m+1} - q^m - q)) \\
&\geq 0, \text{ when } m \geq 3.
\end{align*}$$

Thus, to summarize for the local FR code under consideration, the conditions of Lemma 17 apply when $m \geq 3$. Thus, we can construct a FR code by consider the disjoint union of $l$ of these local FR codes using Construction 2. The code will be optimal with respect to the bound derived in Corollary 4 for file sizes of the form $tq^m + q^{m-1}$ for $1 \leq t < l$.

**Example 15:** A projective plane of order $q$ also forms a FR code $C = (\Omega, V)$, where $\alpha = q + 1$ and $\rho = q + 1$. Furthermore, $|V_i \cap V_j| = 1$ if $i \neq j$ and each pair of symbols appears in exactly one node; this further implies that $\beta = 1$. A simple counting argument shows that $|\Omega| = \theta = q^2 + q + 1$ and $n = q^2 + q + 1$. The value of $\Delta$ (cf. Definition 2) can be determined in the following manner. Applying Corradi’s Lemma, we note that any $q^2 + 1$ nodes cover at least a number of symbols greater than or equal to

$$\frac{(q + 1)^2(q^2 + 1)}{q^2 + q + 1} = q^2 + q + \frac{q + 1}{q^2 + q + 1} > q^2 + q,$$

whereby we conclude that $q^2 + 1$ nodes cover all the $q^2 + q + 1$ symbols. It can also be observed that there is a set of $q^2$ nodes that do not cover all the $q^2 + q + 1$ symbols as the repetition degree of the symbols is $q + 1$. Thus, in this case we can observe that $\Delta = q^2$.

We construct a locally recoverable FR code $\tilde{C}$ by taking $l > 1$ copies of the code $C$. So the code $\tilde{C}$ has parameters $(l(q^2 + q + 1), l(q^2 + q + 1), q + 1, q + 1)$. Let the file size be $M = t(q^2 + q + 1) + q + 1$ for some $1 \leq t < l$. Then, $\tilde{C}$ is optimal with respect to Lemma 16 and has $\rho_{res} = q$. An example is illustrated in Fig. 12.

VI. CONCLUSIONS AND FUTURE WORK

In this work we have constructed several classes of fractional repetition codes that can be used in distributed storage systems. These codes allow for a repair process that is exact and uncoded but table-based. Our constructions
Fig. 12. The figure shows a DSS where \(n = 28\), \(k = 15\), \(r = 3\), \(\theta = 28\), \(\alpha = 3\), \(\rho = 3\) and each local FR code (the columns in the figure) is a projective plane of order 2 which is also known as a Fano plane. Here \(\rho_{res} - 1 = \rho_{loc} = 2\). Any set of 15 nodes cover at least \(M = 17\) symbols. Thus, the minimum distance of the code is 14 for the file size \(M = 17\).

stem from combinatorial designs such as Steiner systems, affine geometries, Hadamard designs and mutually orthogonal Latin squares. We demonstrate that (i) the repetition degree of the symbols which dictates the failure resilience of the code can be varied in an easy manner, and (ii) instances of codes with \(\beta > 1\) that cannot be obtained in a trivial manner from codes with \(\beta = 1\). In addition, we show that new FR codes can be obtained from taking Kronecker products of existing ones and analyze their properties. For codes with exact, uncoded and local repair property (where \(d < k\)), we establish an appropriate minimum distance bound and present constructions from high-girth graphs and collections of local FR codes (with specific properties) that meet these bounds. For most of our constructions, we determine the code rate for specific ranges of \(k\).

There are several opportunities for future work. In principle, several combinatorial design can be treated as FR codes. However, it would be interesting to examine if there are other families that have desirable properties and lend themselves to an analysis of the system code rate. It is to be noted that the code rate depends on the minimum size of the union of \(k\)-sized subsets of the storage nodes. It can also be viewed as determining the expansion level of a bipartite graph derived from the incidence matrix of the design. In general, it is somewhat challenging as most results in the literature only discuss pairwise intersections. A related problem would be determine feasible and infeasible parameter ranges for FR codes.

VII. APPENDIX

Proof of Lemma 4

Note that the properties of net FR codes imply that any two storage nodes intersect in either one or zero symbols. Thus, \(\alpha k - \binom{k}{2}\) is the lower bound on the file size. In the discussion below we demonstrate the existence of \(k\) nodes that cover exactly \(\alpha k - \binom{k}{2}\) symbols. Let the parallel classes be indexed from 0 to \(\rho - 1\).

1: Choose a node \(V_0\) from 0-th parallel class. Initialize \(H = \emptyset, S = \{V_0\}\) and \(i = 1\).
2: while $|H| \leq a$ and $|S| < k$ do
3: Choose $V_i$ from the $i$-th parallel class such that $V_i \cap V_i \notin H$ for all $V_i \in S$.
4: Set $H = H \cup \cup_{V_i \in S} V_i \cap V_i$.
5: Set $S = S \cup V_i$.
6: end while

We need to show that an appropriate $V_i$ can always be chosen in the algorithm and that $|S| = k$ upon exit. To see this note that $H$ tracks the set of pairwise intersections between the nodes at all times. At the beginning of stage $i$, the size of $H$ is at most $\binom{i}{2}$ (by interpreting $\binom{i}{2} = 0$). Note that a parallel class has $a$ nodes and that two nodes from the same parallel class do not intersect. Thus, as long as $a > \binom{i}{2}$ we can always find an appropriate $V_i$. By our assumption $\binom{k-1}{2} < a$. Thus, the algorithm exits with $|S| = k$.

Lemma 18: Consider sets $A_1, \ldots, A_k$ such that $|A_i| = \alpha$ and $|A_i \cap A_j| \leq 1$ when $i \neq j$ and $|\cup_{i=1}^k A_i| = k\alpha - \binom{k}{2}$. This implies that $|A_i \cap A_j| = 1$ for $i \neq j$ and $|A_i \cap A_j \cap A_l| = 0$ for all distinct triples $(i, j, l)$ where $i, j, l = 1, \ldots, k$.

Proof: By the inclusion exclusion principle, we have that

$$|\cup_{i=1}^k A_i| \geq \sum_i |A_i| - \sum_{i<j} |A_i \cap A_j| \geq k\alpha - \binom{k}{2}$$

However, as $|\cup_{i=1}^k A_i| = k\alpha - \binom{k}{2}$, this implies that $|A_i \cap A_j| = 1$ for all pairs $(i, j)$ such that $i \neq j$.

For a set $I \subseteq [k]$, let $A_I$ denote the set $\cap_{i \in I} A_i$. We note that the given conditions also imply that

$$\sum_{\phi \neq I \subseteq [k], |I| \geq 3} (-1)^{|I|+1} |A_I| = 0 \tag{7}$$

We argue that it has to be the case that $|A_I| = 0$ for $\phi \neq I \subseteq [k], |I| \geq 3$. Suppose that this is not the case and there are $l$ subsets $I_1, \ldots, I_l$ such that $|I_i| \geq 3, i = 1, \ldots, l$ and $|A_{I_i}| = 1$. For each $I_i$, there has to be a maximal $I_i^*$ such that $I_i \subset I_i^*$. Moreover, it has to hold that $|I_i^* \cap I_j^*| \leq 1$, as otherwise $I_i^* \cup I_j^*$ provides an example of a subset that is larger than both $I_i^*$ and $I_j^*$. This establishes that for each $I_i$, there is a unique maximal $I_i^*$.

Now, we examine contribution of each of the identified maximal subsets $I_i^*$ to the LHS of eq. (7). It is evident that $|A_I| = 1$ for all $\phi \neq J \subseteq I_i^*$. Let $|I_i^*| = \delta$. This implies that the subset $I_i^*$ induces the following contribution to the LHS of eq. (7): $\sum_{i=3}^\delta (-1)^{i+1} \binom{\delta}{2} - (\delta - 1) > 0$. Thus, the subset $I_i^*$ of maximum cardinality contributes a net positive value to the LHS of eq. (7). Following this we can repeat this argument on the next maximal subset. Note that as the maximal subsets have an intersection of size at most one, each maximal subset contributes the LHS of eq. (7) via distinct terms. Finally, it can be observed that the overall contribution of the maximal subsets accounts for all terms in the LHS of eq. (7). We conclude that if there exist $|A_I| > 0$ for $\phi \neq I \subseteq [k], |I| \geq 3$, we have $\sum_{\phi \neq I \subseteq [k], |I| \geq 3} (-1)^{|I|+1} |A_I| > 0$, which is a contradiction.

\begin{flushright}
\textbf{\textsuperscript{\textcopyright}}
\end{flushright}

References

[1] Wikipedia, “List of device bit rates, available at [http://en.wikipedia.org/wiki/list of device bandwidths].”
[2] A. G. Dimakis, P. B. Godfrey, Y. Wu, M. J. Wainwright, and K. Ramchandran, “Network coding for distributed storage systems,” IEEE Trans. on Info. Th., vol. 56, no. 9, pp. 4539–4551, 2010.
[3] P. Gopalan, C. Huang, H. Simitci, and S. Yekhanin, “On the locality of codeword symbols,” IEEE Trans. on Info. Th., vol. 58, no. 11, pp. 6925–6934, 2012.
[4] D. S. Papailiopoulos and A. G. Dimakis, “Locally repairable codes,” in IEEE Intl. Symposium on Info. Th., 2012, pp. 2771 –2775.
[5] F. Oggier and A. Datta, “Self-repairing homomorphic codes for distributed storage systems,” in Proceedings IEEE INFOCOM, 2011, pp. 1215–1223.

[6] S. Jickak, A.-M. Kermarrec, N. L. Scouarnec, G. Straub, and A. V. Kempen, “Regenerating codes: A system perspective,” ACM SIGOPS Operating Systems Review, vol. 47, no. 2, pp. 23–32, 2013.

[7] S. E. Rouayheb and K. Ramchandran, “Fractional repetition codes for repair in distributed storage systems,” in 48th Annual Allerton Conference on Communication, Control, and Computing, 2010, pp. 1510–1517.

[8] D. R. Stinson, Combinatorial designs: construction and analysis. Springer, 2004.

[9] K. V. Rashmi, N. B. Shah, P. V. Kumar, and K. Ramchandran, “Explicit construction of optimal exact regenerating codes for distributed storage,” in 47th Annual Allerton Conference on Communication, Control, and Computing, 2009, pp. 1243–1249.

[10] K. V. Rashmi, N. B. Shah, and P. V. Kumar, “Optimal exact-regenerating codes for distributed storage at the MSR and MBR points via a product-matrix construction,” IEEE Trans. on Info. Th., vol. 57, no. 8, pp. 5227–5239, 2011.

[11] C. Suh and K. Ramchandran, “Exact-repair mds code construction using interference alignment,” IEEE Trans. on Info. Th., vol. 57, no. 3, pp. 1425–1442, 2011.

[12] C. Tian, V. Aggarwal, and V. A. Vaishampayan, “Exact-repair regenerating codes via layered erasure correction and block designs,” in IEEE Intl. Symposium on Info. Th., 2013, pp. 1431–1435.

[13] 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. on Info. Th., vol. 58, no. 4, pp. 2134–2158, 2012.

[14] I. Tamo, Z. Wang, and J. Bruck, “Mds array codes with optimal rebuilding,” in IEEE Intl. Symposium on Info. Th., 2011, pp. 1240–1244.

[15] O. Olmez and A. Ramamoorthy, “Repairable replication-based storage systems using resolvable designs,” in 50th Annual Allerton Conference on Communication, Control, and Computing (Allerton), 2012, pp. 1174–1181.

[16] D. S. Papailiopoulos, A. G. Dimakis, and V. R. Cadambe, “Repair optimal erasure codes through hadamard designs,” in 49th Annual Allerton Conference on Communication, Control, and Computing, 2011, pp. 1382–1389.

[17] 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. on Info. Th., vol. 58, no. 3, pp. 1837–1852, March 2012.

[18] G. M. Kamath, N. Prakash, V. Lalitha, and P. V. Kumar, “Codes with local regeneration,” in Information Theory and Applications Workshop (ITA), 2013, pp. 1–5.

[19] A. S. Rawat, O. O. Koyluoglu, N. Silberstein, and S. Vishwanath, “Optimal locally repairable and secure codes for distributed storage systems,” IEEE Trans. on Info. Th., vol. 60, no. 1, pp. 212–236, 2014.

[20] O. Olmez and A. Ramamoorthy, “Replication based storage systems with local repair,” in International Symposium on Network Coding (NetCod), 2013, pp. 1–6.

[21] K. W. Shum and Y. Hu, “Functional-repair-by-transfer regenerating codes,” in IEEE Intl. Symposium on Info. Th., July 2012, pp. 1192–1196.

[22] Y. Hu, P. P. C. Lee, and K. W. Shum, “Analysis and construction of functional regenerating codes with uncoded repair for distributed storage systems,” in Proceedings IEEE INFOCOM, 2013, pp. 2355–2363.

[23] J. C. Koo and J. T. Gill, “Scalable constructions of fractional repetition codes in distributed storage systems,” in 49th Annual Allerton Conference on Communication, Control, and Computing, 2011, pp. 1366–1373.

[24] T. Ernvall, “The existence of fractional repetition codes,” 2012, [Online] Available: http://arxiv.org/abs/1201.3547

[25] N. Silberstein and T. Etzion, “Optimal fractional repetition codes,” 2014, [Online] Available: http://arxiv.org/abs/1401.4734

[26] M. Greig and A. Rosa, “Maximal arcs in steiner systems s(2, 4, v),” Discrete Mathematics, vol. 267, no. 1, pp. 143–151, 2003.

[27] R. A. Horn and C. R. Johnson, Matrix Analysis. Cambridge University Press, 2012.

[28] F. Yates, “A new method of arranging variety trials involving a large number of varieties,” The Journal of Agricultural Science, vol. 26, no. 03, pp. 424–455, 1936.

[29] R. C. Bose, S. S. Shrikhande, and E. T. Parker, “Further results on the construction of mutually orthogonal latin squares and the falsity of euler’s conjecture,” Canad. J. Math, vol. 12, pp. 189–203, 1960.
[34] C. W. Lam, L. Thiel, and S. Swiercz, “The non-existence of finite projective planes of order 10,” *Canad. J. Math*, vol. 41, no. 6, pp. 1117–1123, 1989.

[35] T. Skolem, “Some Remarks on the Triple Systems of Steiner.” *Mathematica Scandinavica*, vol. 6, pp. 273–280, 1958.

[36] G. Quattrocchi and H. Zeitler, “Hyperovals in steiner triple systems,” *Journal of Geometry*, vol. 47, no. 1, pp. 125–130, 1993.

[37] H. Sachs, “Regular graphs with given girth and restricted circuits,” *Journal of the London Mathematical Society*, vol. 1, no. 1, pp. 423–429, 1963.

[38] F. Lazebnik and V. A. Ustimenko, “Explicit construction of graphs with an arbitrary large girth and of large size,” *Discrete Applied Mathematics*, vol. 60, no. 1, pp. 275–284, 1995.

[39] S. Jukna, *Extremal combinatorics*. Springer, 2001.