Maximally Recoverable Codes: Connections to Generic Network Coding and Maximal Matching

Chi Wan Sung
Dept. of Electronic Engineering
City University of Hong Kong
Kowloon, Hong Kong
albert.sung@cityu.edu.hk

Kenneth W. Shum
Institute of Network Coding
The Chinese Univ. of Hong Kong
Shatin, N.T., Hong Kong
wkshum@inc.cuhk.edu.hk

Quan Yu
School of Information Engineering
Wuhan University of Technology,
Wuhan, China
yuquan@whut.edu.cn

Guangping Xu
School of Computer & Commun. Eng.
Tianjin University of Technology,
Tianjin, China
xugp@tjut.edu.cn

***Abstract*—The instantiation of a maximally recoverable (MR) code is shown to be a special case of generic network coding. The defining condition of MR codes, called potential independence, is shown to be equivalent to maximal matching in bipartite graphs. Algorithms for MR instantiation are proposed and upper bounds on the required field size are derived.**

I. Introduction

Traditionally, codes for data storage are designed mainly for reliability. For modern distributed storage systems (DSS), there is an increasing concern on efficient repairing of failed storage nodes. To reduce disk I/O during the repair process, a new family of erasure codes called *locally repairable (LR)* codes has been introduced independently in [1] and [2]. The key concept of LR codes is symbol locality, denoted by r, which means that a lost symbol of a codeword can be recovered from no more than r other symbols of the same codeword. Some constructions of LR codes can be found in [3], [4], [5].

Most existing works related to LR codes implicitly assume that all storage nodes are simply fully connected with one another. Network topology is not taken into account for code design. In practical DSS, storage nodes may be separated geographically and connected through a network, which may also consists of intermediate devices such as switches or routers. To address the design issues of practical DSS, our previous work [6] focuses on the design of LR codes over a heterogeneous network with an arbitrary topology. A related work can be found in [7].

Our work in [6] considers only binary codes. Using larger field size, the erasure-correction capability of the code can be enhanced. Given a code topology, some erasure patterns are intrinsically uncorrectable. For potentially correctable erasure patterns, it is most desirable if all of them can be corrected by assigning field elements properly subject to code topology constraints. The resultant code is said to be *maximally recoverable (MR)*, a concept first discussed in [8] and subsequently studied in [9], [10]. In this paper, we investigate this concept by connecting it to generic network coding and maximal bipartite matching. Our contributions are three-fold. First, the problem of MR instantiation is shown to be a generic network coding problem with a four-layer network topology. Second, the concept of potential independence is shown to be equivalent to maximal matching in bipartite graphs. Third, two algorithms for MR instantiation are designed and the corresponding upper bounds on the required field size are derived.

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

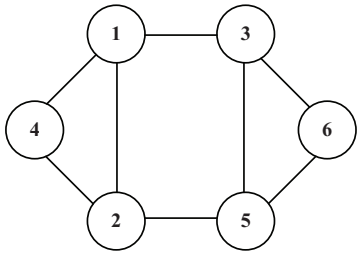

Fig. 1. A storage network $\mathcal{G}$ with six nodes.

II. Motivation

Consider a storage network $\mathcal{G}$, which consists of six storage nodes labeled from 1 to 6, as shown in Fig. 1. We assume that the repair groups have been chosen for this network, for example, by the algorithm in [6]. Suppose that they are $R_1 = \{1, 2, 4\}, R_2 = \{1, 2, 3, 5\}$, and $R_3 = \{3, 5, 6\}$. Let $\mathcal{C}$ be a linear LR code over GF(q) for $\mathcal{G}$ and $\boldsymbol{c} = (c_1, c_2, \ldots, c_6)$ be a codeword of $\mathcal{C}$. Since a failed node, by definition, can be repaired by all other nodes together in the same repair group, we can obtain the following three equations:

$$\begin{aligned} h_{11}c_1 + h_{12}c_2 + h_{14}c_4 &= 0, \\ h_{21}c_1 + h_{22}c_2 + h_{23}c_3 + h_{25}c_5 &= 0, \\ h_{33}c_3 + h_{35}c_5 + h_{36}c_6 &= 0, \end{aligned}$$

where $h_{i,j}$'s are elements of GF(q). Let

$$\boldsymbol{H} = \begin{bmatrix} h_{11} & h_{12} & 0 & h_{14} & 0 & 0 \\ h_{21} & h_{22} & h_{23} & 0 & h_{25} & 0 \\ 0 & 0 & h_{33} & 0 & h_{35} & h_{36} \end{bmatrix} \quad (1)$$

be the parity-check matrix for $\mathcal{C}$. It can be seen that some components of $\boldsymbol{H}$ are restricted to be zero due to the code topology while others are variables to be determined. Note that if we remove the columns $3, 5, 6$ of $\boldsymbol{H}$, the resultant sub-matrix is intrinsically singular, which means that the erasure pattern $\{3, 5, 6\}$ are *information-theoretically uncorrectable*, no matter how large the field size is and how we assign the variables.

Similarly, the erasure pattern $\{1,2,4\}$ are also information-theoretically uncorrectable. Except for the erasure patterns $\{1,2,4\}$ and $\{3,5,6\}$, other erasure patterns containing three storage nodes are *potentially correctable*. For example, if the ternary field is used and the variables are assigned such that

$$\boldsymbol{H} = \begin{bmatrix} 2 & 2 & 0 & 1 & 0 & 0 \\ 2 & 1 & 2 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix},$$

we can check that all erasure patterns of size 3 are correctable except for the erasure patterns $\{1,2,4\}$ and $\{3,5,6\}$. From this example, we can see that if the code topology is fixed, the capability of erasure tolerance can be maximized if the coefficients of parity-check equations are properly chosen from a sufficiently large finite field. It is noteworthy that if the field size is not large enough, potentially correctable erasure patterns may not all be made correctable no matter how we choose the coefficients. For example, if the binary field is used instead, for $\boldsymbol{H}$ given in (1), there is no way to correct all erasure patterns in the form of $\{j,4,6\}$ for $j \in \{1,2,3,5\}$.

III. Problem Formulation

Consider an (n,k) linear code $\mathcal{C}$ over GF(q), where q is the field size. Let $\boldsymbol{H}$ be an $(n-k)\times n$ parity-check matrix for $\mathcal{C}$. Some components of $\boldsymbol{H}$ are restricted to be zero due to network topology. Others are variables to be determined. To avoid degenerate cases, we assume that each column of $\boldsymbol{H}$ contains at least one variable. An assignment of values to these variables is called an *instantiation* of $\boldsymbol{H}$. A set of columns of $\boldsymbol{H}$ is said to be *potentially independent* if there exists a field over which there is an instantiation of $\boldsymbol{H}$ that makes the set of columns linearly independent.

Definition 1. *An instantiation of $\boldsymbol{H}$ is said to be maximally recoverable (MR) if every potentially independent set of columns of $\boldsymbol{H}$ is linearly independent under that instantiation.*

It has been proved in [9, Lemma 32] that MR instantiation always exists, provided that the field size is large enough.

Theorem 1. *An erasure pattern with erasures in locations indexed by $\mathcal{J} \subseteq \{1,2,\ldots,n\}$ is correctable if and only if the columns of $\boldsymbol{H}$ with indices in $\mathcal{J}$ are linearly independent.*

Proof. We know that the columns of $\boldsymbol{H}$ with indices in $\mathcal{J}$ are linearly independent if and only if there is no nonzero codeword with support in $\mathcal{J}$.

Suppose that the columns of $\boldsymbol{H}$ indexed by $\mathcal{J}$ are linearly independent. If the erasure pattern corresponding to $\mathcal{J}$ is not correctable, then there must be at least two codewords whose restrictions to the complement of $\mathcal{J}$ are identical. The difference of these two codewords is a nonzero codeword with support in $\mathcal{J}$, which is a contradiction.

Conversely, suppose the columns of $\boldsymbol{H}$ indexed by $\mathcal{J}$ are linearly dependent, so that there is a nonzero codeword with support in $\mathcal{J}$. We cannot distinguish this nonzero codeword with the all-zero codeword if the symbols indexed by $\mathcal{J}$ are erased. Therefore, it is an uncorrectable erasure pattern. □

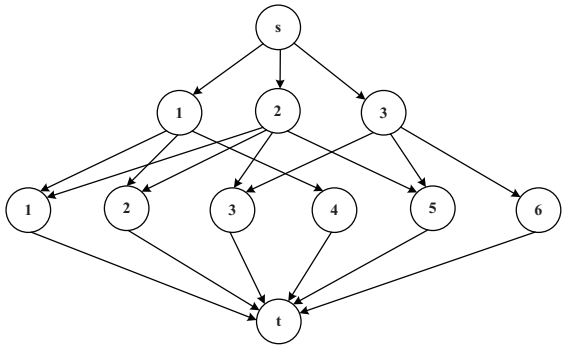

Fig. 2. $\mathcal{G}_{\boldsymbol{H}}$ for the parity-check matrix $\boldsymbol{H}$ in (1).

The above result implies the following equivalent definition:

Definition 2. *An instantiation of $\boldsymbol{H}$ is said to be MR if the resultant code can correct every erasure pattern that is correctable by some instantiation of $\boldsymbol{H}$.*

IV. Connection to Generic Network Coding

In this section, we show that an MR instantiation of a given $\boldsymbol{H}$ is equivalent to a generic network code for a corresponding single-source acyclic network. Given $\boldsymbol{H}$, we define an acyclic network that has four layers of nodes as follows. In the top layer, there is a single source node s. In the second layer, there are $\omega \triangleq n-k$ nodes, each of which is connected by a directed edge from s. In the third layer, there are n nodes. If the (i,j)-th entry of $\boldsymbol{H}$ is a variable, then there is a directed edge from node i in the second layer to node j in the third layer. In the bottom layer, there is only one single node t, and there is a directed edge from each node in the third layer to node t. We denote the network by $\mathcal{G}_{\boldsymbol{H}}$. The edges in $\mathcal{G}_{\boldsymbol{H}}$ can also be naturally classified into three layers, namely, the top layer, the middle layer and the bottom layer. The network corresponding to the parity-check matrix given in (1) is presented in Fig. 2.

Consider an ω-dim linear network code for such a network. This can be interpreted as that the source node s has ω symbols to transmit, each of which belongs to a finite field. To represent these symbols, one may consider that there are ω *imaginary edges* that has no originating nodes but terminate at s. These edges have the natural basis of the ω-dimensional space as their *global encoding kernels*. They are vectors in the ω-dimensional space, which represent how the messages are encoded. Without loss of generality, we assume that s only forwards the symbols without performing any coding operations, so the global encoding kernels of its ω outgoing edges are just the natural basis. Each node in the second layer has one and only one incoming edge, so it can only forward the incoming symbol. Therefore, the global encoding kernels of its outgoing edges are all the same and equal to the global encoding kernel of its incoming edge. In general, a node in the third layer can have more than one incoming edges. Since linear network codes are considered, it can linearly combine its incoming symbols, which means that the global encoding kernel of an outgoing edge can be expressed as a linear

combination of the global encoding kernels of its incoming edges. Edges in the *top* and *middle* layers that have the same global encoding kernels are said to be in an *equivalence class*. Since coding is performed only by nodes in the third layer, for $i = 1, 2, \ldots, n$, we let $\boldsymbol{f}_i$ be the global encoding kernel of the outgoing edge of node i in the third layer.

In an acyclic graph, two paths are said to be *edge-disjoint* if they do not share a common edge. Given any collection ξ of edges, let V_ξ be the linear span of the global encoding kernels of the edges in ξ, and maxflow(ξ) be the maximum number of edge-disjoint paths which terminate at the edges in ξ.

Definition 3. *An ω-dim linear network code is said to be generic if for any non-empty collection ξ of edges, the condition*

$$|\xi| = \min\{\omega, \textit{maxflow}(\xi)\} \tag{2}$$

implies dim$(V_\xi) = |\xi|$.

More details of generic network codes can be found in [11, Chapter 19]. Due to the topology of the network $\mathcal{G}_{\boldsymbol{H}}$, the condition for a network code to be generic can be simplified, so that it suffices to consider only collection ξ of edges in the *bottom* layer, as shown in the following result:

Theorem 2. *An ω-dim linear network code for $\mathcal{G}_{\boldsymbol{H}}$ is generic if for any non-empty collection ξ of edges in the bottom layer, the condition*

$$|\xi| = \textit{maxflow}(\xi) \tag{3}$$

implies dim$(V_\xi) = |\xi|$.

Proof. First, notice that there are only ω outgoing edges of node s. Given any collection ξ of edges, we must have

$$\min\{\omega, \text{maxflow}(\xi)\} = \text{maxflow}(\xi),$$

so condition (2) can be simplified to (3).

Next, consider the case where ξ consists of edges *not* in the bottom layer and $|\xi| = \text{maxflow}(\xi)$. Denote one such edge by e. Since $|\xi| = \text{maxflow}(\xi)$, all other edges in the equivalence class of e do not belong to ξ, for otherwise the number of edge-disjoint paths from s to ξ must be smaller than $|\xi|$. By construction, the global encoding kernel $\boldsymbol{f}_e$ of e has a non-zero element in one and only one coordinate. Define $\sigma \triangleq \xi \backslash \{e\}$ and U_σ as the projection of V_ξ onto the orthogonal complement of $\boldsymbol{f}_e$. It is clear that $|\sigma| = \text{maxflow}(\sigma)$. Moreover, if $\dim(U_\sigma) = |\sigma|$, then $\dim(V_\xi) = |\xi|$. By repeating the argument, it suffices to consider only the case where ξ consists of only edges in the bottom layer. □

By the construction of $\mathcal{G}_{\boldsymbol{H}}$, it is clear that a network code for $\mathcal{G}_{\boldsymbol{H}}$ is an instantiation of $\boldsymbol{H}$, and vice versa. Our aim is to prove that an instantiation being *maximally recoverable* is equivalent to a network code being *generic*. Before doing this, we need to establish the relationship between the potential independency of columns of $\boldsymbol{H}$ and the maxflow of an edge collection of $\mathcal{G}_{\boldsymbol{H}}$:

Theorem 3. *A set of columns of $\boldsymbol{H}$ is potentially independent if and only if the corresponding collection ξ of edges in the bottom layer of $\mathcal{G}_{\boldsymbol{H}}$ satisfies condition* (3)*.*

Proof. Consider a collection ξ of edges in the bottom layer that satisfies (3). Since the number of edge-disjoint paths from s to ξ is equal to the number of edges in ξ, s can simply send a set of $|\xi|$ linearly independent vectors over these paths, and thus the global encoding kernels of these edges are linearly independent. If these global encoding kernels are regarded as an instantiation of $\boldsymbol{H}$, then the corresponding columns are linearly independent. Hence, the condition (3) implies that the corresponding columns of $\boldsymbol{H}$ are potentially independent.

Next consider a set of columns of $\boldsymbol{H}$ that are potentially independent. Let the corresponding collection of edges in the bottom layer of $\mathcal{G}_{\boldsymbol{H}}$ be ξ. We are going to prove by contradiction that (3) holds. By definition of maxflow, we must have $|\xi| \geq \text{maxflow}(\xi)$, since the number of edge-disjoint paths terminating at edges in ξ cannot be larger than $|\xi|$. Suppose strict inequality holds, i.e.,

$$|\xi| > \text{maxflow}(\xi). \tag{4}$$

Construct a new graph $\tilde{\mathcal{G}}$ from $\mathcal{G}_{\boldsymbol{H}}$ by removing all edges in the bottom layer of $\mathcal{G}_{\boldsymbol{H}}$ that do not belong to ξ. It is clear that maxflow(ξ) in the original graph is equal to the number of edge-disjoint paths from s to t in $\tilde{\mathcal{G}}$. We denote this number by N. By the edge-connectivity version of Menger's Theorem (e.g. [12, Theorem 28.5]), the minimum cut for s and t in $\tilde{\mathcal{G}}$ contains N edges. Since the set of columns of $\boldsymbol{H}$ are potentially independent, there exists an instantiation such that these column vectors are linearly independent. Equivalently, there exists a network code such that

$$|\xi| = \dim(V_\xi). \tag{5}$$

On the other hand, we must have $\dim(V_\xi) \leq N$, since the global encoding kernels of these N edges, which together form a cut between s and t, must lie within an N-dimensional subspace. By the hypothesis in (4), $N < |\xi|$. Combining the two inequalities, we have $\dim(V_\xi) \leq N < |\xi|$, which contradicts with (5). Hence, a set of columns of $\boldsymbol{H}$ being potentially independent implies that the corresponding edge collection ξ satisfies (3). □

We are now ready to prove the main result of this section:

Theorem 4. *An instantiation of $\boldsymbol{H}$ with $\boldsymbol{f}_1, \ldots, \boldsymbol{f}_n$ as its columns is an MR instantiation if and only if $\boldsymbol{f}_1, \ldots \boldsymbol{f}_n$ form an ω-dim generic network code for the network $\mathcal{G}_{\boldsymbol{H}}$, where $\omega \triangleq n - k$.*

Proof. Consider a set of columns of $\boldsymbol{H}$ that is potentially independent. Let the corresponding edge collection be ξ. By Theorem 3, condition (3) holds. If $\boldsymbol{f}_1, \ldots, \boldsymbol{f}_n$ form a generic network code, by Theorem 2, $\dim(V_\xi) = |\xi|$. Therefore, those columns of $\boldsymbol{H}$ are linearly independent. Since this is true for all potentially independent set of columns, the corresponding instantiation is MR. This proves the backward part.

Next, consider an MR instantiation. If an edge collection ξ satisfies (3), then by Theorem 3, the corresponding columns are potentially independent. Since the instantiation is MR, the global encoding kernels of the edges in ξ are indeed linearly independent, implying that $\dim(V_\xi) = |\xi|$. Hence, the corresponding network code is generic, which completes the forward part. □

V. Maximal Bipartite Matching

In the last section, we have constructed a four-layer network based on $\boldsymbol{H}$. Suppose we remove from $\boldsymbol{H}$ the source node s, the sink node t, and all edges incident to them. Then we obtain a bipartite graph. We denote it by $\mathcal{G}'_{\boldsymbol{H}}(\mathcal{X}, \mathcal{Y})$, where $\mathcal{X}$ is the index set of nodes in the second layer while $\mathcal{Y}$ is the index set of nodes in the third layer.

Definition 4. *A set $\mathcal{M}$ of edges in a graph is called a matching if no two edges in $\mathcal{M}$ have a vertex in common. Moreover, $\mathcal{M}$ is said to be a maximal matching if it is not a proper subset of another matching.*

Theorem 5. *A set of columns of $\boldsymbol{H}$, indexed by $\mathcal{I}$, is potentially independent if and only if there is a matching $\mathcal{M}$ of $\mathcal{G}'_{\boldsymbol{H}}(\mathcal{X}, \mathcal{Y})$ so that the nodes in $\mathcal{Y}$ that are indexed by $\mathcal{I}$ are all incident by edges in $\mathcal{M}$.*

Proof. This is a direct consequence of Theorem 3. Observe that $|\mathcal{I}| = |\xi|$, and condition (3) is equivalent to that there are $|\xi|$ edge-disjoint paths terminating at ξ. These edge-disjoint paths, removing those edges in the top and bottom layer of $\mathcal{G}_{\boldsymbol{H}}$, are the matching required. □

Recall that an MR instantiation requires that all potentially independent columns of $\boldsymbol{H}$ are linearly independent. Clearly, it suffices to consider only the sets of columns that are *maximally* potentially independent, which means that adding an extra column to such a set will violate the condition of potential independence. Given $\boldsymbol{H}$, denote the collection of the index sets of maximally potentially independent columns by $\Omega_C(\boldsymbol{H})$, and the collection of all maximal matchings in $\mathcal{G}'_{\boldsymbol{H}}(\mathcal{X}, \mathcal{Y})$ by $\Omega_M(\boldsymbol{H})$[1]. Theorem 5 implies that we can define a surjective mapping f from Ω_M to Ω_C. We remark that the members of Ω_M can be enumerated by the algorithm in [13], and the mapping f can be explicitly found.

In general, f is not injective. We define the partial inverse of f, denoted by f^{-1}, by restricting the domain. In other words, f^{-1} maps $\mathcal{I} \in \Omega_c$ to $\mathcal{M} \in \Omega_M$ such that $f(\mathcal{M}) = \mathcal{I}$. Note that if more than one such maximal matchings exist, $f^{-1}(\mathcal{C})$ can be defined as any of them.

By construction, each edge in the bipartite graph $\mathcal{G}'_{\boldsymbol{H}}(\mathcal{X}, \mathcal{Y})$ corresponds to a variable in $\boldsymbol{H}$. Given any $\mathcal{I} \in \Omega_C$, the maximal matching $\mathcal{M}_\mathcal{I} \triangleq f^{-1}(\mathcal{I})$ consists of $|\mathcal{I}|$ edges, which connects an $|\mathcal{I}|$-subset of $\mathcal{X}$, denoted by $\mathcal{X}_\mathcal{I}$, to $\mathcal{I} \subset \mathcal{Y}$. Define $\boldsymbol{H}(\mathcal{M}_\mathcal{I})$ as the square submatrix of $\boldsymbol{H}$ by preserving the rows indexed by $\mathcal{X}_\mathcal{I}$ and the columns indexed by $\mathcal{I}$.

[1]When there is no ambiguity, we may omit their dependence on $\boldsymbol{H}$.

Theorem 6. *An instantiation of $\boldsymbol{H}$ is an MR instantiation if*

$$\prod_{\mathcal{I} \in \Omega_C} \det\left(\boldsymbol{H}(\mathcal{M}_\mathcal{I})\right) \neq 0. \tag{6}$$

Proof. The condition implies that every determinant in (6) is non-zero, and $\boldsymbol{H}(\mathcal{M}_\mathcal{I})$ is of full rank for $\mathcal{I} \in \Omega_C$. That means, all maximally potentially independent columns are linearly independent, and the instantiation is MR by definition. □

Corollary 7. *An MR instantiation of $\boldsymbol{H}$ exists if the field size, q, satisfies*

$$q > \max_x N_x(\boldsymbol{H}), \tag{7}$$

where the maximum is taken over all variable x in $\boldsymbol{H}$ and

$$N_x(\boldsymbol{H}) \triangleq |\{\mathcal{I} \in \Omega_c(\boldsymbol{H}) : \textit{column index of } x \in \mathcal{I}\}|. \tag{8}$$

In addition, (7) *holds if*

$$q > \sum_{i=1}^{\omega} \binom{n-1}{i-1}. \tag{9}$$

Proof. The condition in (6) can be expressed as a multi-variate polynomial inequation. Consider a variable x which occurs in $\boldsymbol{H}(\mathcal{M}_\mathcal{I})$. Regarding other variables as fixed, $\det\left(\boldsymbol{H}(\mathcal{M}_\mathcal{I})\right)$ is linear in x. As a result, the highest possible degree of x in (6) is $N_x(\boldsymbol{H})$. If (7) holds, then q is larger than the degree of every variable in (6). The first statement then follows directly from Schwartz-Zippel Lemma [11, Lemma 19.17].

To prove the second statement, observe that the number of $\mathcal{I}$ in (8) with $|\mathcal{I}| = i$ is at most $\binom{n-1}{i-1}$. N_x can then be bounded above by summing $\binom{n-1}{i-1}$ for i from 1 to ω. □

VI. MR Instantiation Algorithms

In this section, we present two MR instantiation algorithms. We add the restriction that each variable must be assigned a *non-zero* element. This constraint is relevant to the design of LR codes, since assigning zero to a variable, in effect, changes the composition of a repair group.

A. Algorithm based on Maximal Bipartite Matching

We design an algorithm that can produce a solution satisfying (6). Let there be v variables in (6), and denote them by $x_1, x_2, \ldots, x_v$. Define $\boldsymbol{x}(l)$ as the vector of the first l variables. Denote the set of all determinants in (6) that contains x_i by $\mathcal{D}_{x_i}$, for $i = 1, 2, \ldots, v$. Since we allow only non-zero values to be assigned, we replace the field size requirement in (9) to

$$q > \sum_{i=1}^{\omega} \binom{n-1}{i-1} + 1. \tag{10}$$

Suppose $\boldsymbol{x}(v-1)$ have been assigned values. Consider the assignment of x_v. Ignoring for a moment those determinants that do not involve x_v, (6) can be expressed as

$$\alpha_d(\boldsymbol{x}(v-1))x_v + \beta_d(\boldsymbol{x}(v-1)) \neq 0 \quad \forall d \in \mathcal{D}_{x_v}, \tag{11}$$

where $\alpha_d(\boldsymbol{x}(v-1))$ is a determinant. If $\alpha_d(\boldsymbol{x}(v-1)) \neq 0$ for all $d \in \mathcal{D}_{x_v}$ and $q - 1 > N_{x_v} \geq |\mathcal{D}_{x_v}|$, then we can always assign a non-zero element in GF(q) to x_v such that (11) holds.

Next consider the assignment of x_{v-1}. For $d \in \mathcal{D}_{x_{v-1}} \cap \mathcal{D}_{x_v}$, we need to ensure that $\alpha_d(\boldsymbol{x}(v-1)) \neq 0$. For $d \in \mathcal{D}_{x_{v-1}} \setminus \mathcal{D}_{x_v}$, we also need to ensure that those determinants are non-zero. Therefore, we need to ensure that $|\mathcal{D}_{x_{v-1}}|$ determinants, in total, are non-zero. By the same argument, this can be done if $\boldsymbol{x}(v-2)$ has been properly assigned and $q-1 > N_{x_{v-1}} \geq |\mathcal{D}_{x_{v-1}}|$. Repeating the argument, an MR instantiation of the v variables can be determined by a recursive procedure, provided that the field size satisfies (10). Other variables which do not occur in (6) can be assigned arbitrarily.

B. Algorithm based on Generic Network Coding

We design an algorithm that can produce a generic network code for a given graph $\mathcal{G}_{\boldsymbol{H}}$. For ease of presentation, we let $X \subset_{<\omega} Y$ to mean that X is a subset of Y and the cardinality of X is less than ω. For $i = 1, 2, \ldots, n$, we denote the i-th edge in the bottom layer of $\mathcal{G}_{\boldsymbol{H}}$ by e_i, which corresponds to the i-th column of $\boldsymbol{H}$. In addition, we denote the vector space spanned by the global encoding kernels of all incoming edges of node i in the third layer by V_i. Given any vector $\boldsymbol{x}$, we use $w(\boldsymbol{x})$ to denote the weight of $\boldsymbol{x}$, which is the number of non-zero components of $\boldsymbol{x}$. To ensure that our proposed algorithm can be successfully executed, we impose the condition

$$q > \sum_{\alpha=0}^{\omega-1} \binom{n-1}{\alpha} + \omega. \tag{12}$$

Algorithm 1 Assign Global Encoding Kernels for Edges in the Bottom Layer of $\mathcal{G}_{\boldsymbol{H}}$

Input: $\mathcal{G}_{\boldsymbol{H}}$, where the dimension of $\boldsymbol{H}$ is $\omega \times n$
Output: $\{\boldsymbol{f}_{e_i} : i = 1, 2, \ldots, n\}$

1: Let $\boldsymbol{f}_{e_1}$ be the unique 0-1 vector in V_1 such that $w(\boldsymbol{f}_{e_1}) = \dim(V_1)$;
2: $U_0 := \{e_1\}$;
3: **for** $i := 2, 3, \ldots, n$ **do**
4: Pick any $\boldsymbol{x} \in V_i$ such that $w(\boldsymbol{x}) = \dim(V_i)$ and $\boldsymbol{x} \notin V_\xi$, where $\xi \subseteq_{<\omega} U_0$ such that $\{\boldsymbol{f}_e \colon e \in \xi\}$ are linearly independent and $V_i \not\subset V_\xi$;
5: $\boldsymbol{f}_{e_i} := \boldsymbol{x}$;
6: $U_0 := U_0 \cup \{e_i\}$;
7: **end for**

Recall that the global encoding kernels of the edges in the middle layer of the network belong to the natural basis. The condition in Line 1, i.e., $w(\boldsymbol{f}_{e_1}) = \dim(V_1)$, implies no zero is assigned in combining the incoming packets. In Line 2, we initialize a set U_0, which stores the global encoding kernels of the edges in the bottom layer that have been determined. The for-loop which starts at Line 3 assigns a global encoding kernel to each of the remaining edges in the bottom layer. The main step is in Line 4, which needs a careful examination.

Note that the vector $\boldsymbol{x}$ in Step 4 can always be found. To see this, we count the number of vectors in V_i that cannot be chosen. Define $\nu \triangleq \dim(V_i)$, and consider a given ξ. Since $V_i \not\subset V_\xi$, we must have $\dim(V_i \cap V_\xi) \leq \nu - 1$. Hence,

$$|V_i \cap V_\xi| \leq q^{\nu-1}. \tag{13}$$

Note that vectors in $V_i \cap V_\xi$ cannot be chosen. In addition, we need to exclude those vectors where $w(\boldsymbol{x}) < \dim(V_i)$. There are $q^\nu - (q-1)^\nu$ such vectors, and it can be shown that

$$q^\nu - (q-1)^\nu < \nu q^{\nu-1}.$$

Therefore, the number of vectors in V_i that cannot be chosen is bounded above by

$$\begin{aligned}
\big|\cup_\xi (V_i \cap V_\xi)\big| + \nu q^{\nu-1} &< \sum_\xi |V_i \cap V_\xi| + \nu q^{\nu-1} && (14)\\
&\leq \sum_\xi q^{\nu-1} + \nu q^{\nu-1} && (15)\\
&\leq \sum_{\alpha=0}^{\omega-1} \binom{n-1}{\alpha} q^{\nu-1} + \nu q^{\nu-1} && (16)\\
&\leq \Big[\sum_{\alpha=0}^{\omega-1} \binom{n-1}{\alpha} + \omega\Big] q^{\nu-1} && (17)\\
&\leq q^\nu = |V_i|, && (18)
\end{aligned}$$

where (14) follows from the union bound, (15) follows from (13), (16) follows from that there are at most $\sum_{\alpha=0}^{\omega-1} \binom{n-1}{\alpha}$ choices of ξ to consider, (17) follows from that $\nu \leq \omega$, and (18) follows from (12). Hence, in Line 4, $\boldsymbol{x}$ can always be found. Following the same argument in [11, p. 467], it can be proved that the output of Algorithm 1 is indeed a generic network code.

REFERENCES

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

[2] F. Oggier and A. Datta, "Self-repairing homomorphic codes for distributed storage systems," in *Proc. IEEE Int. Conf. on Computer Commun. (INFOCOM '11)*, Shanghai, China, Apr. 2011, pp. 1215–1223.

[3] D. S. Papailiopoulos and A. G. Dimakis, "Locally repairable codes," in *Proc. IEEE Int. Symp. Inf. Theory*, St. Petersburg, Russia, Aug. 2012, pp. 2771–2775.

[4] W. Song, S. H. Dau, C. Yuen, and J. Li, "Optimal locally repairable linear codes," *IEEE J. on Selected Areas in Commun.*, vol. 32, no. 5, pp. 1019–1036, 2014.

[5] A. S. Rawat, O. O. Koyluoglu, N. Silberstein, and S. Vishwanath, "Optimal locally repairable and secure codes for distributed storage systems," *IEEE Trans. Inf. Theory*, vol. 60, no. 1, pp. 212–236, 2014.

[6] Q. Yu, C. W. Sung, and T. H. Chan, "Locally repairable codes over a network," in *Proc. IEEE Information Theory Workshop (ITW)*, Hobart, Tasmania, Australia, Nov. 2014, pp. 70–74.

[7] A. Mazumdar, "On a duality between recoverable distributed storage and index coding," in *Proc. IEEE Int. Symp. Inf. Theory*, Honolulu, HI, USA, Jul. 2014, pp. 1977–1981.

[8] M. Chen, C. Huang, and J. Li, "On maximally recoverable property for multi-protection group codes," in *Proc. IEEE Int. Symp. Inf. Theory*, Nice, France, Jun. 2007, pp. 486–490.

[9] P. Gopalan, C. Huang, B. Jenkins, and S. Yekhanin, "Explicit maximally recoverable codes with locality," *IEEE Trans. Inf. Theory*, vol. 60, no. 9, pp. 5245–5256, 2014.

[10] P. Gopalan, G. Hu, and S. Kopparty, "Maximally recoverable codes for grid-like topology," in *Proc. ACM-SIAM Symp. Discrete Alg. (SODA)*, Barcelona, Spain, Jan. 2017, pp. 2092–2108.

[11] R. W. Yeung, *Information Theory and Network Coding*. Springer, 2008.

[12] R. J. Wilson, *Introduction to Graph Theory*, 5th ed. Pearson, 2012.

[13] T. Uno, "Algorithms for enumerating all perfect, maximum and maximal matchings in bipartite graphs," in *Lecture Notes in Computer Science*. Springer, 1997, vol. 1350.