# A General Coded Caching Scheme for Scalar Linear Function Retrieval

Yinbin Ma and Daniela Tuninetti

***Abstract*—Coded caching aims to minimize the network's peak-time communication load by leveraging the information pre-stored in the local caches at the users. The original setting by Maddah-Ali and Niesen, which considered single file retrieval, has been recently extended to general Scalar Linear Function Retrieval (SLFR) by Wan *et al.*, who proposed a linear scheme that surprisingly achieves the same optimal load under the constraint of uncoded cache placement as in single file retrieval. This paper's goal is to characterize the conditions under which a general SLFR linear scheme is optimal and gain insights into why the specific choices made by Wan *et al.* work. This paper shows that the optimal decoding coefficients are necessarily the product of two terms, one only involving the encoding coefficients and the other only the demands of the users. In addition, the algebraic relationships among the encoding coefficients of an optimal code are shown to be captured by the cycles of a *universal graph*. Thus, a general linear scheme for the SLFR problem can be found by solving a spanning tree problem for the universal graph. The proposed framework readily extends to caching-like problems, such as the problem of finding a general linear scheme for Sun *et al.*'s private function computation.**

***Index Terms*—Coded caching; Scalar Linear Function Retrieval; Universal graph; Spanning tree;**

## I. Introduction

Coded caching, originally introduced by Maddah-Ali and Niesen (MAN) in [2], has been the focus of much research efforts recently as it predicts, for networks with a server delivering a single file to each cache-aided user, that it is possible to achieve a communication load that does not scale with the number of users. A coded caching scheme, with one server, $\mathsf{K}$ users and $\mathsf{N}$ files, comprises two phases. During the *placement phase*, the server distributes content in the $\mathsf{K}$ caches as a function of the $\mathsf{N}$ files stored in its library but still ignoring the user demands. During the *delivery phase*, the server broadcasts to the users a signal through a shared link; the signal is function of the library, the cached contents and the demands. The goal is to minimize the numbers of information bits sent through the shared link during the delivery phase for the worst-case demands. The MAN achievable scheme includes an uncoded cache placement phase and a (linear network) coded delivery phase, which comprising several *multicast messages*. Each multicast message benefits a subset of $t{+}1$ users simultaneously, where $t$ is a parameter that relates to the aggregate amount of cache memory in the system; therefore, the MAN scheme is able to reduce the worst-case communication load by a factor $t+1$ compared to a scheme that serves the demand of each user one by one. The MAN scheme is not optimal. Yu *et al.* (YMA) in [3] improved on the delivery phase of the MAN scheme by removing those multicast messages that are linearly dependent on others, which may happen when a file is requested by multiple users and the cache size is small. The YMA scheme in [3] meets the converse bound first proposed by Wan *et al.* in [4] derived under the constraint of *uncoded cache placement*. When the assumption of uncoded cache placement is relaxed (i.e., coding is allowed also in the placement phase), Yu *et al.* in [3] showed that the YMA scheme has at most twice the load of the yet-to-be-found optimal coded caching scheme.

Much work followed the original MAN paper [2]. Some works focused on achievable schemes with coded placement [5] while others on tightening MAN's original cut-set bound [6]. Another line of development has considered extensions of the MAN's original setting. For example, coded caching with secure delivery was studied in [7], with private demands in [8], with more realistic network topologies in [9], [10], with Device-to-Device delivery in [11], etc. Ideas from coded caching have found applications to other problems, such as coded distributed computing [12], coded data shuffling [13]–[15], and private information retrieval [16], [17].

Directly relevant to this work, is Wan *et al.*'s extension of the MAN setup so as to allow users to request general scalar linear combinations of the files stored in the library [18]. Despite the fact that the number of possible demands in Scalar Linear Function Retrieval (SLFR) increases exponentially in the number of files compared to MAN's single file retrieval, Wan *et al.* in  [18] surprisingly showed that the optimal communication load remains the same in both settings, at least under the constraint of uncoded cache placement.

The scheme proposed in [18] uses a linear code in the delivery phase, in which the server selects a set of *leader users* (i.e., whose demand vectors are a linearly independent spanning set of the set of all possible demands) and creates multicast messages by performing linear combinations of the demanded subfiles that were not cached. The coefficients for such linear combinations are referred to as *encoding coefficients* and can be optimized. As in the YMA scheme, multicast messages that would only be useful for non-leader users are not sent and have to be locally reconstructed as a linear combination of the sent multicast messages; the coefficients for such linear combinations are referred to as *decoding coefficients* and must guarantee that each non-leader user correctly decodes its demanded linear combination of files. The choice of encoding and decoding coefficients in [18] is rather non trivial and not a simple extension of the MAN scheme, which actually fails to guarantee successful decoding

The authors are with the Electrical and Computer Engineering Department of the University of Illinois Chicago, Chicago, IL 60607, USA (e-mail: yma52@uic.edu, danielat@uic.edu). Part of this work was presented at the 2021 IEEE International Symposium on Information Theory (ISIT) [1]. This work was supported in part by NSF Award 1910309.

on finite fields of characteristics strictly larger than two. The encoding coefficients chosen in [18] were inspired by Private Function Retrieval (PFR) [17]: they all have unit modulo but alternate in sign between by the leader users' and non-leader users' demands. The corresponding decoding coefficients in [18] are, up to a sign, the determinants of certain matrices derived from the demand matrix. Why such a PFR-inspired choice works could not be explained in [18] (and neither in [17], for that matter).

### A. Paper Contribution

This paper aims to gain fundamental insights into why the choices in [18] (and in [17], for that matter) work by analyzing the most general YMA-like linear scheme (i.e., general encoding and decoding coefficients). Our main contribution is to show that the optimal decoding coefficients are necessarily the product of two terms, one only involving the encoding coefficients and the other only the determinants of certain matrices derived from the demand matrix. In addition, we characterize the algebraic relationships which the encoding coefficients need to satisfy in order to guarantee successful decoding at all users as the cycles of a certain graph, which we name *universal graph* (because it does not depend on the demand matrix). Thus, we show that a general YMA-like scheme for SLFR, which is optimal under uncoded placement, can be found by solving a spanning tree problem.

### B. Paper Organization

The rest of the paper is organized as follow. Section II introduces the SLFR problem and summarizes relevant results. Section III presents our main result: a general YMA-like linear scheme for SLFR can be found by solving a spanning tree problem on a universal graph, whose proof can be found in Sections IV and V. Section VI shows how the proposed framework extends to PFR. Section VII concludes the paper. Proofs of some auxiliary results can be found in Appendix.

### C. Notation Convention

In this paper we use the following notation convention. Sans-serif symbols denote system parameters. $|\cdot|$ denotes either the cardinality of a set or the length of a vector. $\det(M)$ is the determinant of the matrix $M$. $1_{\{\mathcal{E}\}}$ is the indicator function of the event $\mathcal{E}$. $M[\mathcal{Q},\mathcal{S}]$ is the sub-matrix of $M$ obtained by selecting the rows indexed by $\mathcal{Q}$ and the columns indexed by $\mathcal{S}$. For an integer $b$, we let $[b] := \{1,\ldots,b\}$. For a ground set $\mathcal{G}$ and an integer $t$, we let $\Omega^t_{\mathcal{G}} := \{\mathcal{T} \subseteq \mathcal{G} : |\mathcal{T}| = t\}$. Moreover, $\mathcal{S} \setminus \mathcal{Q} := \{k : k \in \mathcal{S}, k \notin \mathcal{Q}\}$. $\mathsf{Ind}_{\mathcal{S},k}$ returns the position of the element $k \in \mathcal{S}$, where the element of the integer set $\mathcal{S}$ are considered in increasing order. For example, $\mathsf{Ind}_{\{3,5\},3} = 1$ and $\mathsf{Ind}_{\{3,5\},5} = 2$. By convention $\mathsf{Ind}_{\mathcal{S},k} = 0$ if $k \notin \mathcal{S}$. $\mathsf{Tot}_{\mathcal{S}}$ returns the sum of all elements in $\mathcal{S}$. For example, $\mathsf{Tot}_{\{2,3\}} = 2 + 3 = 5$. By convention $\mathsf{Tot}_{\emptyset} = 0$.

## II. Problem Formulation and Known Results

### A. Problem Formulation

A $(\mathsf{K},\mathsf{N},\mathsf{q})$ scalar linear function retrieval (SLFR) problem is defined as follows. A central server stores a library of $\mathsf{N}$ files, where each file has $\mathsf{B}$ independent and uniformly distributed symbols over the finite field $\mathbb{F}_{\mathsf{q}}$, for some prime-power $\mathsf{q}$. Files are denoted as $F_1,\ldots,F_{\mathsf{N}}$. The server communicates through an error-free shared link to $\mathsf{K}$ users. User $k \in [\mathsf{K}]$ has a local memory denoted as $Z_k \in \mathbb{F}_{\mathsf{q}}^{\mathsf{MB}}$, for some $\mathsf{M} \in [0,\mathsf{N}]$. $\mathsf{M} \in [0,\mathsf{N}]$ is referred to as the *memory* size, measured in multiple of the file size. The SLFR problem includes two phases, *cache placement* phase and *delivery* phase.

*a) Cache Placement Phase:* the server populates the caches with content from the library files, that is

$$H(Z_k|F_1,\ldots F_{\mathsf{N}}) = 0,\ \forall k \in [\mathsf{K}]. \tag{1}$$

During this phase, the server is unaware of what content the users will request in the future.

*b) Delivery Phase:* User $k \in [\mathsf{K}]$ demands a scalar linear function of the files denoted as

$$B_k := d_{k,1}F_1 + \ldots + d_{k,\mathsf{N}}F_{\mathsf{N}} \in \mathbb{F}_{\mathsf{q}}^{\mathsf{B}},\ \forall k \in [\mathsf{K}], \tag{2}$$

where the row vector $\mathbf{d}_k = (d_{k,1},\ldots,d_{k,\mathsf{N}}) \in \mathbb{F}_{\mathsf{q}}^{\mathsf{N}}$ is referred to as the *demand vector of user k*. The demand matrix $\mathbb{D} := [\mathbf{d}_1;\ldots;\mathbf{d}_{\mathsf{K}};] \in \mathbb{F}_{\mathsf{q}}^{\mathsf{K}\times\mathsf{N}}$ collects the demands of all the users. Once the server receives $\mathbb{D}$, it generates a message $X \in \mathbb{F}_{\mathsf{q}}^{\mathsf{RB}}$ from the files, that is

$$H(X|F_1,\ldots,F_{\mathsf{N}},\mathbb{D}) = 0. \tag{3}$$

The server broadcasts the signal $X$ to all users. $\mathsf{R} \in [0,\min\{\mathsf{K},\mathsf{N}\}]$ is referred to as the communication *load*, measured in multiple of the file size.

*c) Decoding:* All users must decode their desired function correctly based on their local cache and the signal sent by the server, that is

$$H(B_k|Z_k,\mathbf{d}_k,X) = 0,\ \forall k \in [\mathsf{K}]. \tag{4}$$

*d) Performance:* Performance is measured as follows.

- Optimal Worst-Case Load: design the cache placement phase in (1) and the delivery phase in (3) that attain

$$\mathsf{R}^\star(\mathsf{M}) = \limsup_{\mathsf{B}\to\infty} \min_{Z_1,\ldots,Z_{\mathsf{K}},X} \max_{\mathbb{D}} \{\mathsf{R} : \text{all above conditions are satisfied}\},\ \forall \mathsf{M} \in [0,\mathsf{N}], \tag{5}$$

  where the maximization over the demand matrix is because we seek to provide performance guarantees for the worst case scenario.
- Optimal Worst-Case Load Under Uncoded Placement: If symbols form the files are directly copied into the caches without coding, the placement phase is said to be *uncoded*. The worst-case load under uncoded placement is denoted as $\mathsf{R}^\star_{\mathsf{u}}(\mathsf{M})$ and is defined as in (5) with the exception that in (1) the placement is restricted to be uncoded. Clearly $\mathsf{R}^\star(\mathsf{M}) \le \mathsf{R}^\star_{\mathsf{u}}(\mathsf{M})$.

### B. General YMA-type SLFR Scheme

For a $(\mathsf{K},\mathsf{N},\mathsf{q})$ SLFR problem, for a fixed $\mathsf{MK}/\mathsf{N} = t \in [0 : \mathsf{K}]$, the following YMA-type scheme is a generalization of the scheme in [18].

*a) Cache Placement Phase:* Partition the set $[\mathsf{B}]$ as

$$[\mathsf{B}] = \left\{ \mathcal{I}_{\mathcal{T}} : \mathcal{I}_{\mathcal{T}} \subseteq [\mathsf{B}], \mathcal{T} \in \Omega^t_{[\mathsf{K}]}, \ |\mathcal{I}_{\mathcal{T}}| = \mathsf{B}/\binom{\mathsf{K}}{t} \right\}, \tag{6}$$

and define (with a Matlab-like notation) the sub-files as

$$F_{i,\mathcal{T}} := F_i(\mathcal{I}_{\mathcal{T}}) \in \mathbb{F}_{\mathsf{q}}^{\mathsf{B}/\binom{\mathsf{K}}{t}}, \ \forall \mathcal{T} \in \Omega^t_{[\mathsf{K}]}, \ \forall i \in [\mathsf{N}]. \tag{7}$$

The cache of user $k \in [\mathsf{K}]$ is populated as

$$Z_k = \{F_{i,\mathcal{T}} : \mathcal{T} \in \Omega^t_{[\mathsf{K}]}, k \in \mathcal{T}, i \in [\mathsf{N}]\} \in \mathbb{F}_{\mathsf{q}}^{\mathsf{BN}\binom{\mathsf{K}-1}{t-1}/\binom{\mathsf{K}}{t}}. \tag{8}$$

The memory size is thus

$$\mathsf{M} = \mathsf{N}\binom{\mathsf{K}-1}{t-1} \Big/ \binom{\mathsf{K}}{t} = \frac{\mathsf{N}t}{\mathsf{K}}. \tag{9}$$

*b) Delivery Phase:* The demands of the users in (2) are represented in the demand matrix $\mathbb{D} := [\mathbf{d}_1; \ldots; \mathbf{d}_{\mathsf{K}};]$. As for the sub-files in (7), we define the *demand-blocks* as

$$B_{k,\mathcal{T}} = B_k(\mathcal{I}_{\mathcal{T}}) \in \mathbb{F}_{\mathsf{q}}^{\mathsf{B}/\binom{\mathsf{K}}{t}}, \ \forall \mathcal{T} \in \Omega^t_{[\mathsf{K}]}, \ \forall k \in [\mathsf{K}], \tag{10}$$

where the demanded scalar linear function $B_k$ was defined in (2). Some demand-blocks can be reconstructed based on the cache content available locally at the users as defined in (8), while the remaining ones need to be delivered by the server.

Define the *set of leader users* $\mathcal{L}$ as

$$\mathcal{L} \subseteq [\mathsf{K}] \ : \ \mathrm{rank}_{\mathsf{q}}(\mathbb{D}) = \mathrm{rank}_{\mathsf{q}}(\mathbb{D}[\mathcal{L}, :]) = |\mathcal{L}|. \tag{11}$$

The set $\mathcal{L}$ is not unique but its size is, as every finite-dimensional vector space has a basis.

Define the *transformed demand matrix* $\mathbb{D}' \in \mathbb{F}_{\mathsf{q}}^{\mathsf{K} \times |\mathcal{L}|}$ as the matrix with entries

$$[\mathbb{D}']_{k,\ell} = \begin{cases} 1_{\{k=\ell\}} & \text{if } k \in \mathcal{L} \\ x_{k,\ell} & \text{if } k \notin \mathcal{L} \end{cases}, \ \forall k \in [\mathsf{K}], \ \forall \ell \in \mathcal{L}, \tag{12}$$

which allows one to express the demand-blocks of non-leader users as a linear combination of the demand-blocks of the leader users as follows

$$B_{k,\mathcal{T}} = \sum_{\ell \in \mathcal{L}} x_{k,\ell} \ B_{\ell,\mathcal{T}}, \ \forall \mathcal{T} \in \Omega^t_{[\mathsf{K}]}, \forall k \in [\mathsf{K}] \setminus \mathcal{L}. \tag{13}$$

The existence of $\{x_{u,\ell} \in \mathbb{F}_{\mathsf{q}} : u \in [\mathsf{K}] \setminus \mathcal{L}, \ \ell \in \mathcal{L}\}$ in (13) follows from a change of basis.

The server forms the following *multicast messages*

$$W_{\mathcal{S}} := \sum_{k \in \mathcal{S}} \alpha_{k,\mathcal{S}\setminus\{k\}} \ B_{k,\mathcal{S}\setminus\{k\}} \in \mathbb{F}_{\mathsf{q}}^{\mathsf{B}/\binom{\mathsf{K}}{t}}, \ \forall \mathcal{S} \in \Omega^{t+1}_{[\mathsf{K}]}, \tag{14}$$

for some *encoding coefficients*

$$\mathrm{ENC} := \{\alpha_{k,\mathcal{S}\setminus\{k\}} \in \mathbb{F}_{\mathsf{q}} \setminus \{0\} : \ \mathcal{S} \in \Omega^{t+1}_{[\mathsf{K}]}, k \in \mathcal{S}\}. \tag{15}$$

The server sends all multicast messages in (14) that are useful for the leader users, that is

$$\begin{aligned} X = \{W_{\mathcal{S}} : \mathcal{S} \in \Omega^{t+1}_{[\mathsf{K}]}, |\mathcal{S} \cap \mathcal{L}| > 0\} \cup \{\mathcal{L}, \mathbb{D}'\} \\ \in \mathbb{F}_{\mathsf{q}}^{\Delta + \mathsf{B}(\binom{\mathsf{K}}{t+1} - \binom{\mathsf{K}-|\mathcal{L}|}{t+1})/\binom{\mathsf{K}}{t}}. \end{aligned} \tag{16}$$

Note that sending the leader set and the transformed demand matrix in (16) requires at most

$$\Delta = |\mathcal{L}|(\lceil \log_{\mathsf{q}}(\mathsf{K}) \rceil + \mathsf{K}) \text{ symbols}, \tag{17}$$

where $\Delta$ in (17) does not scale with the file length $\mathsf{B}$.

From (16) (we shall see in the next subsections that successful decoding is possible with this transmitted signal), we see that the worst-case load occurs when the leader set size is the largest possible, that is when $|\mathcal{L}| = \min(\mathsf{K}, \mathsf{N})$, and the resulting load is

$$\mathsf{R} = \left[ \binom{\mathsf{K}}{t+1} - \binom{\mathsf{K} - \min(\mathsf{K},\mathsf{N})}{t+1} \right] \Big/ \binom{\mathsf{K}}{t}. \tag{18}$$

*c) Decoding:* Successful decoding is possible with the transmitted signal in (16) and the cache contents in (8) if the following holds. For a given $\mathcal{S} \in \Omega^{t+1}_{[\mathsf{K}]}$, user $k \in \mathcal{S}$ can decode its desired demand-block $B_{k,\mathcal{S}\setminus\{k\}}$ from $W_{\mathcal{S}}$ by "caching out" all the other blocks in $W_{\mathcal{S}}$. Thus, by construction of (16), every leader user can retrieve its demanded scalar linear function. For the non-leader users, we need to show how to locally reconstruct the non-sent multicast messages

$$X_{\text{not-sent}} = \{W_{\mathcal{A}} : \mathcal{A} \in \Omega^{t+1}_{[\mathsf{K}]\setminus\mathcal{L}}\}, \tag{19}$$

from the transmitted signal in (16); if so, each non-leader user can retrieve its demanded scalar linear function.

For $\mathsf{K} - |\mathcal{L}| \geq t+1$, the local reconstruction of $X_{\text{non-sent}}$ in (19) is possible if we can express

$$W_{\mathcal{A}} = \sum_{\mathcal{S} \in \Omega^{t+1}_{[\mathsf{K}]}, |\mathcal{S} \cap \mathcal{L}| > 0} \beta^{(\mathcal{A})}_{\mathcal{S}} \ W_{\mathcal{S}}, \ \forall \mathcal{A} \in \Omega^{t+1}_{[\mathsf{K}]\setminus\mathcal{L}}, \tag{20}$$

by finding an appropriate set of *decoding coefficients*

$$\begin{aligned} \mathrm{DEC} := \{\beta^{(\mathcal{A})}_{\mathcal{S}} \in \mathbb{F}_{\mathsf{q}} : \mathcal{S} \in \Omega^{t+1}_{[\mathsf{K}]}, \\ |\mathcal{S} \cap \mathcal{L}| > 0, \ \mathcal{A} \in \Omega^{t+1}_{[\mathsf{K}]\setminus\mathcal{L}}\}. \end{aligned} \tag{21}$$

At this point the question is to determine which choice of encoding coefficients in the linear scheme in (14) ensures we can satisfy (20), where the choice of encoding coefficients in (15) and of decoding coefficients in (21) must work for all realizations of the demands and all realizations of the files[1]. Next we give the solution proposed in [18].

### C. Known Results from [18]

In [18] it was shown that successful decoding is possible if one alternates between $+1$ and $-1$ the encoding coefficients in (14) as follows

$$\alpha_{k,\mathcal{S}\setminus\{k\}} = (-1)^{\mathsf{Ind}_{\mathcal{S}\cap\mathcal{L},k} + \mathsf{Ind}_{\mathcal{S}\setminus\mathcal{L},k}}, \tag{22}$$

and the resulting decoding coefficients are

$$\beta^{(\mathcal{A})}_{\mathcal{S}} = (-1)^{1+\mathsf{Tot}_{\mathcal{A}\setminus\mathcal{S}}} \cdot \det\left(\mathbb{D}'[\mathcal{A} \setminus \mathcal{S}, \ \mathcal{S} \setminus \mathcal{A}]\right), \tag{23}$$

where the functions $\mathsf{Ind}$ in (22) and $\mathsf{Tot}$ in (23) were defined in the Section I-C. Therefore one concludes the following.

**Theorem 1** (Results from [18])**.** *For a* $(\mathsf{K}, \mathsf{N}, \mathsf{q})$ *SLFR problem, requesting arbitrary scalar linear functions of the files from the server does not incur any load penalty with respect to requesting a single file, i.e., the lower convex envelope of the*

[1] The leader set in (11), the encoding coefficients in (15) and the decoding coefficients in (21) are all function of the demand matrix $\mathbb{D}$. Such a dependency is not made explicit here in order not to clutter the notation.

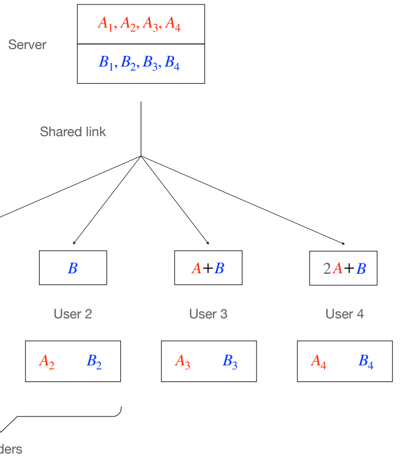


Fig. 1: MAN cache placement and SLFR demands.

*following points from [3] is achievable for the SLFR problem as well [18]: for all* $t \in [0 : \mathsf{K}]$

$$(\mathsf{M}, \mathsf{R}_{\text{YMA}}) = \left( \frac{\mathsf{N}t}{\mathsf{K}}, \frac{\binom{\mathsf{K}}{t+1} - \binom{\mathsf{K}-\min(\mathsf{N},\mathsf{K})}{t+1}}{\binom{\mathsf{K}}{t}} \right). \tag{24}$$

*In addition, as for the single file retrieval problem,* $\mathsf{R}^\star_{\mathsf{u}} = \mathsf{R}_{\text{YMA}}$ *[3], [18], and* $\mathsf{R}^\star_{\mathsf{u}} \leq 2\mathsf{R}^\star$ *[6].* □

### *D. Example of SLFR system with parameters* $(\mathsf{N}, \mathsf{K}, |\mathcal{L}|, t) = (2, 4, 2, 1)$*: scheme in [18]*

We realize that the notation used in the description of the achievable scheme in the previous subsections may be difficult to follow. We thus provide here a detailed example to illustrate the scheme in Section II-C from [18]. We consider the SLFR problem with $\mathsf{N} = 2$ files, $\mathsf{K} = 4$ users, $|\mathcal{L}| = 2$ leaders, and memory parameter $t = 1$.

*a) Cache Placement Phase:* As shown in Fig. 1, the caches are populated as in the MAN scheme.

*b) Delivery Phase:* We consider the user demands

$$\mathbb{D} = \mathbb{D}' = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 1 \\ 2 & 1 \end{bmatrix}. \tag{25}$$

In this example the leader users are indexed by $\mathcal{L} = \{1, 2\}$. In response to the demands in (25), the SLFR scheme in [18] constructs the following multicast messages (see choice of encoding coefficients in (22))

| $\mathcal{S}$ | $W_{\mathcal{S}}$ | | |
|---|---|---|---|
| $\{1,2\}$ | $A_2$ | $-$ | $B_1$ |
| $\{1,3\}$ | $A_3$ | $+$ | $(A_1 + B_1)$ |
| $\{1,4\}$ | $A_4$ | $+$ | $(2A_1 + B_1)$ |
| $\{2,3\}$ | $B_3$ | $+$ | $(A_2 + B_2)$ |
| $\{2,4\}$ | $B_4$ | $+$ | $(2A_2 + B_2)$ |
| $\{3,4\}$ | $(A_4 + B_4)$ | $-$ | $(2A_3 + B_3)$ |

(26)

in which all encoding coefficients are equal to $+1$ with the only exception of a negative sign for $W_{\{1,2\}}$ and for $W_{\{3,4\}}$, because those are the only multicast messages that benefit more than one leader user or more than one non-leader user. The server sends all multicast messages in (26) except $W_{\{3,4\}}$, because $W_{\{3,4\}}$ is only useful to non-leader users.

*c) Decoding:* The leader users can decode their missing demand-blocks from the sent multicast messages. The non-leader users need $W_{\{3,4\}}$ to recover a demand-block not computable from their local cache content, i.e., $(A_4 + B_4)$ is needed by user 3, and $(2A_3 + B_3)$ by user 4.

According to the SLFR scheme in [18], the non-leader users locally reconstruct $W_{\{3,4\}}$ by taking a linear combination of the multicast messages in (26), except $W_{\{3,4\}}$, with decoding coefficients (see choice of decoding coefficients in (23)),

| $\mathcal{S}$ | $\beta_{\mathcal{S}}^{(\{3,4\})}$ | |
|---|---|---|
| $\{1,2\}$ | $(-1)^{1+(3+4)} \det(\mathbb{D}[\{3,4\},\{1,2\}])$ | $= -1$ |
| $\{1,3\}$ | $(-1)^{1+4} \det(\mathbb{D}[\{4\},\{1\}])$ | $= -2$ |
| $\{1,4\}$ | $(-1)^{1+3} \det(\mathbb{D}[\{3\},\{1\}])$ | $= +1$ |
| $\{2,3\}$ | $(-1)^{1+4} \det(\mathbb{D}[\{4\},\{2\}])$ | $= -1$ |
| $\{2,4\}$ | $(-1)^{1+3} \det(\mathbb{D}[\{3\},\{2\}])$ | $= +1$ |

We can verify that as shows Fig. 2 Therefore the non-leader users can recover their missing demand-block. The attained memory-load pair is $(\mathsf{M}, \mathsf{R}) = (2/4, 5/4)$, which coincides with (24) for the choice of parameters in this example.

In the next section we describe the most general version of this linear YMA-type SLFR scheme.

## III. Main Results

The Authors of [18] did not provide a fundamental reason as of why the choice of alternating signs in (22) for the encoding coefficients (and the resulting decoding coefficients in (23)) allows successful decoding by all users; they mentioned they were inspired by PFR [17]. The open question we aim to answer in this paper is whether the choice in [18] is fundamental, that is, whether (up to scaling) the only choice of encoding and decoding coefficients that guarantees successful decoding in any YMA-type scheme is the one found in [18] given by (22) and (23). We answer this open question by analyzing the most general linear scheme in the form of (14) and (20). In a nutshell, our main contribution is to show that:

1) the signs of the encoding coefficients must follow a pattern where they alternate in sign (but not necessarily as in (22)) and their modulo need not be one;
2) the decoding coefficients are proportional to the determinants of certain matrices obtained from the transformed demand matrix as in (23), but the proportionality coefficient need not have modulo one; and
3) finally and importantly, the encoding and decoding coefficients must satisfy certain relationships that are captured by the cycles of certain *universal graph*. **Thus, we show that a general SLFR linear scheme can be found by solving a spanning tree problem.**

### *A. Main Result 1: General YMA-type SLFR Scheme*

Our main result is to show that the linear scheme in Section II-B guarantees successful decoding under the conditions stated in Theorem 2. We provide the proof of Theorem 2 in Section IV.

| | $-W_{\{1,2\}}$ | $-2W_{\{1,3\}}$ | $+W_{\{1,4\}}$ | $-W_{\{2,3\}}$ | $+W_{\{2,4\}}$ | $= W_{\{3,4\}}$ |
|---|---|---|---|---|---|---|
| $A_1$ | | $-2$ | $+2$ | | | |
| $A_2$ | $-1$ | | | $-1$ | $+2$ | |
| $A_3$ | | $-2$ | | | | $-2$ |
| $A_4$ | | | $+1$ | | | $+1$ |
| $B_1$ | $+1$ | $-2$ | $+1$ | | | |
| $B_2$ | | | | $-1$ | $+1$ | |
| $B_3$ | | | | $-1$ | | $-1$ |
| $B_4$ | | | | | $+1$ | $+1$ |

Fig. 2: Local reconstruction of $W_{\{3,4\}}$ by exploiting the decoding coefficients.

**Theorem 2.** *For the* $(\mathsf{K},\mathsf{N},\mathsf{q})$ *SLFR problem with memory size parameterized by* $t$ *as in* (9)*, given the demand matrix* $\mathbb{D}$*, fix the of set of leader users* $\mathcal{L}$ *as in* (11) *and determine the corresponding transformed demand matrix* $\mathbb{D}'$ *as in* (12)*. For each* $\mathcal{A} \in \Omega^{t+1}_{[\mathsf{K}]\setminus\mathcal{L}}$*, the decoding coefficients defined in* (21) *for the non-sent multicast message* $W_{\mathcal{A}}$ *must be of the form*

$$\beta^{(\mathcal{A})}_{\mathcal{S}} = \widetilde{\beta}^{(\mathcal{A})}_{\mathcal{S}} \cdot \det\left(\mathbb{D}'[\mathcal{A} \setminus \mathcal{S},\ \mathcal{S} \setminus \mathcal{A}]\right),\ \forall \mathcal{S} \in \Omega^{t+1}_{\mathcal{A}\cup\mathcal{L}}, \quad (27a)$$

*where* $\widetilde{\beta}^{(\mathcal{A})}_{\mathcal{S}}$ *in* (27a) *only depends on the encoding coefficients in* (15) *through the relationship*

$$\widetilde{\beta}^{(\mathcal{A})}_{\mathcal{S}} \cdot \alpha_{k,\mathcal{T}} \cdot (-1)^{\phi^{(\mathcal{A})}_{k,\mathcal{T}}} = \mathsf{c}^{(\mathcal{A})}_{\mathcal{T}},$$
$$\forall k \in \mathcal{S},\ \text{where } \mathcal{T} = \mathcal{S} \setminus \{k\}, \quad (27b)$$

*for some set of constants* $\{\mathsf{c}^{(\mathcal{A})}_{\mathcal{T}} \in \mathbb{F}_{\mathsf{q}} : \mathcal{T} \in \Omega^{t}_{\mathcal{A}\cup\mathcal{L}}\}$ *and where the 'sign function'* $\phi^{(\mathcal{A})}_{k,\mathcal{T}}$ *in* (27b) *is given by*

$$\phi^{(\mathcal{A})}_{k,\mathcal{T}} = \begin{cases} 1 + \mathsf{Ind}_{(\{k\}\cup\mathcal{T})\setminus\mathcal{A},k} & k \in \mathcal{L}\setminus\mathcal{T} \\ \mathsf{Ind}_{\mathcal{A}\setminus\mathcal{T},k} & k \in \mathcal{A}\setminus\mathcal{T} \end{cases},$$
$$\forall k \in \mathcal{S},\ \text{where } \mathcal{T} = \mathcal{S} \setminus \{k\}, \quad (27c)$$

*where the function* $\mathsf{Ind}$ *was defined in Section I-C.* □

**Remark 1.** Our main result in Theorem 2 generalizes the scheme described in Section II-C from [18]. The reason why the signs of the encoding coefficients (and the resulting decoding coefficients) must alternate in [18] as given in (22) is because of the 'sign function' in (27c), which is satisfied by the choice in (22). However, the alternating patten in (22) is just one possible feasible linear scheme in Theorem 2. The choice of encoding coefficients in (22) (and the resulting decoding coefficients in (23)) has the following advantages: (a) the scheme does not involve divisions other than by elements of unit modulo, which in turns allows one to extend the scheme to monomial retrieval as well, as noted in [18]; and (b) the scheme works irrespective of the characteristics of the finite field. □

### B. Example of SLFR system with parameters $(\mathsf{N},\mathsf{K},|\mathcal{L}|,t) = (2,4,2,1)$: general scheme

We consider the same SLFR problem as in Section II-D. The placement is the same as shown in Fig. 1. We consider the user demands in (25). The most general version of the multicast messages in (26) is

| $\mathcal{S}$ | $W_{\mathcal{S}}$ | | |
|---|---|---|---|
| $\{1,2\}$ | $\alpha_{1,\{2\}}A_2$ | $+$ | $\alpha_{2,\{1\}}B_1$ |
| $\{1,3\}$ | $\alpha_{1,\{3\}}A_3$ | $+$ | $\alpha_{3,\{1\}}(A_1+B_1)$ |
| $\{1,4\}$ | $\alpha_{1,\{4\}}A_4$ | $+$ | $\alpha_{4,\{1\}}(2A_1+B_1)$ |
| $\{2,3\}$ | $\alpha_{2,\{3\}}B_3$ | $+$ | $\alpha_{3,\{2\}}(A_2+B_2)$ |
| $\{2,4\}$ | $\alpha_{2,\{4\}}B_4$ | $+$ | $\alpha_{4,\{2\}}(2A_2+B_2)$ |
| $\{3,4\}$ | $\alpha_{3,\{4\}}(A_4+B_4)$ | $+$ | $\alpha_{4,\{3\}}(2A_3+B_3)$ |

(28)

and the non-sent multicast message $W_{\{3,4\}}$ is reconstructed locally by a linear combination of the other multicast messages by using decoding coefficients

| $\mathcal{S}$ | $\beta^{(\{3,4\})}_{\mathcal{S}}$ |
|---|---|
| $\{1,2\}$ | $\frac{\alpha_{3,\{4\}}\alpha_{4,\{2\}}}{\alpha_{1,\{2\}}\alpha_{2,\{4\}}}\det\left(\mathbb{D}[\{3,4\},\{1,2\}]\right)$ |
| $\{1,3\}$ | $\frac{\alpha_{4,\{3\}}}{\alpha_{1,\{3\}}}\det\left(\mathbb{D}[\{4\},\{1\}]\right)$ |
| $\{1,4\}$ | $\frac{\alpha_{3,\{4\}}}{\alpha_{1,\{4\}}}\det\left(\mathbb{D}[\{3\},\{1\}]\right)$ |
| $\{2,3\}$ | $\frac{\alpha_{4,\{3\}}}{\alpha_{2,\{3\}}}\det\left(\mathbb{D}[\{4\},\{2\}]\right)$ |
| $\{2,4\}$ | $\frac{\alpha_{3,\{4\}}}{\alpha_{2,\{4\}}}\det\left(\mathbb{D}[\{3\},\{2\}]\right)$ |

(29)

How to obtain the decoding coefficients in (29) from the relationships in (27) is explained in the next subsection. More details on this specific example can be found in Section V-A as well; we defer this detailed derivation after we have introduced some concepts in Section V.

### C. Main Result 2: Universal Graph

The relationship in (27b) (i.e., a relationship that must be satisfied by the part of the decoding coefficients that does not depend on the demand matrix) can be represented on a graph. This graph representation is useful to find a set of encoding coefficients that satisfy Theorem 2 as clarified next in Proposition 1.

**Proposition 1.** *For a fixed non-sent multicast message* $W_{\mathcal{A}}$ *(i.e., such that* $\mathcal{A} \in \Omega^{t+1}_{[\mathsf{K}]\setminus\mathcal{L}}$*), the conditions identified by Theorem 2 can be represented on an undirected bipartite graph that has the* $\widetilde{\beta}^{(\mathcal{A})}_{\mathcal{S}}$*'s and the* $\mathsf{c}^{(\mathcal{A})}_{\mathcal{T}}$*'s appearing in* (27b) *as vertices, and whose edges are labeled by the encoding coefficients according to the relationship in* (27b)*. An unweighted spanning tree on such a graph identifies all the encoding coefficients that are free to vary, in other words, cycles on such a graph identify constraints that the encoding coefficients must satisfy.* □

**Remark 2.** Proposition 1 identifies a graph for each non-sent multicast message, that is, for each subset $\mathcal{A} \in \Omega^{t+1}_{[\mathsf{K}]\setminus\mathcal{L}}$; we shall thus refer to the graph for multicast message $W_{\mathcal{A}}$ as graph-$\mathcal{A}$. All these graphs share edges, by which we mean the encoding coefficients that label the edges. This is so because the vertices are unique to each graph-$\mathcal{A}$ (i.e., their superscript identify which subset $\mathcal{A}$ they refer to) but the labels on the edges are not. The labels on the edges are in fact, up to a sign, the encoding coefficients. The same encoding coefficient appears in many graphs if strictly more than one non-sent multicast message must be locally reconstructed by the non-leader users (see for example Fig. 4). This presents a problem when we need to simultaneously find a spanning tree in more than one graph because we need to make sure that the encoding coefficients that are not part of a spanning tree in one graph (i.e., that are not free to vary) retain the same value in the other graphs; we refer to this condition as "consistency among graphs.". In the conference version [1] of this journal paper we presented a greedy algorithm to ensure consistency among graphs. In this journal version we present a different approach to the problem of consistency among graphs: we seek a *universal solution* for the $(\widetilde{\beta}^{(\mathcal{A})}_{\{k\}\cup\mathcal{T}}, \mathsf{c}^{(\mathcal{A})}_{\mathcal{T}})$'s that that satisfy Theorem 2 and that does not depend on $\mathcal{A}$; this is accomplished by obtaining a novel 'sign function' instead of (27c) that does not depend on $\mathcal{A}$. As a result, all the relationships captured by the graph-$\mathcal{A}$'s can be expressed in a single *universal graph*. By finding a spanning tree for the universal graph, we thus determine all encoding coefficients that are free to vary at once, resulting in Proposition 2 whose proof is in Section V. □

**Proposition 2.** *Theorem 2 has the following as a solution. All parameters* $(\widetilde{\beta}^{(\mathcal{A})}_{\{k\}\cup\mathcal{T}}, \mathsf{c}^{(\mathcal{A})}_{\mathcal{T}})$*'s in* (27) *do not depend on* $\mathcal{A}$ *and the 'sign function' in* (27c) *can be equivalently replaced by*

$$\phi'_{k,\mathcal{T}} := \begin{cases} \mathsf{Ind}_{\{k\}\cup\mathcal{T}\setminus\overline{\mathcal{L}},k} & k \in \mathcal{L} \\ \mathsf{Ind}_{\overline{\mathcal{L}}\setminus\mathcal{T},k} & k \in \overline{\mathcal{L}} \end{cases}, \ \forall \mathcal{T} \in \Omega^{t}_{[\mathsf{K}]}, \ \overline{\mathcal{L}} := [\mathsf{K}] \setminus \mathcal{L}. \tag{30}$$

**Remark 3.** Theorem 2 and Proposition 2 are derived from the SLFR caching problem. In Section VI we shall see that the derived framework applies to the various download sub-phases of the PFR scheme proposed in [17]. □

## IV. Proof of Theorem 2

We shall start to prove the result for the case $\mathsf{K}-|\mathcal{L}| = t+1$ in Section IV-A (i.e., only the multicast message indexed by $\mathcal{A} = [\mathsf{K}] \setminus \mathcal{L}$ must be reconstructed in (20)). Then, in Section IV-B, we shall argue that the case $\mathsf{K}-|\mathcal{L}| > t+1$ can be solved by analyzing several "reduced systems" with only $|\mathcal{L}|+t+1$ users in each system, for all $|\mathcal{L}| \in [\min(\mathsf{K},\mathsf{N})]$ and $t \in [0:\mathsf{K}]$. In the following, for a subset $\mathcal{T}$ of $[\mathsf{K}]$, we let $\overline{\mathcal{T}} := [\mathsf{K}] \setminus \mathcal{T}$.

### A. Case $\mathsf{K}-|\mathcal{L}| = t+1$

We consider here a system with $\mathsf{K}$ users, $|\mathcal{L}| = r$ (where $r$ stands for "rank") leader users, and memory size parameterized by $t$, where $(t,r)$ are fixed and satisfy $\mathsf{K} = r+t+1$. In particular, $\overline{\mathcal{L}}$ is the set of non-leader users.

Given $\mathcal{L}$, define the transformed demand matrix as in (12). Here only the multicast message indexed by $\mathcal{A} = \overline{\mathcal{L}}$ needs to be reconstructed, thus for notation convenience we drop $\mathcal{A}$ from $\beta^{(\mathcal{A})}_{\mathcal{S}}$ in (20) so as not to clutter the notion; thus $\mathcal{A}\cup\mathcal{L} = [\mathsf{K}]$. We re-write the condition in (20) with $\beta_{\overline{\mathcal{L}}} = -1$ as follow

$$\sum_{\mathcal{S}\in\Omega^{t+1}_{[\mathsf{K}]}} \beta_{\mathcal{S}} W_{\mathcal{S}} \tag{31a}$$

$$= \sum_{\mathcal{S}\in\Omega^{t+1}_{[\mathsf{K}]}} \beta_{\mathcal{S}} \sum_{k\in\mathcal{S}} \alpha_{k,\mathcal{S}\setminus\{k\}} \sum_{\ell\in\mathcal{L}} [\mathbb{D}']_{k,\ell} B_{\ell,\mathcal{S}\setminus\{k\}} \tag{31b}$$

$$= \sum_{\mathcal{T}\in\Omega^{t}_{[\mathsf{K}]}} \sum_{\ell\in\mathcal{L}} \sum_{k\in\overline{\mathcal{T}}} \beta_{\{k\}\cup\mathcal{T}} \ \alpha_{k,\mathcal{T}} \ [\mathbb{D}']_{k,\ell} B_{\ell,\mathcal{T}} \tag{31c}$$

$$= 0 \in \mathbb{F}_{\mathsf{q}}^{\mathsf{B}/\binom{\mathsf{K}}{t}}, \tag{31d}$$

where (31b) is because of the definition of multicast messages in (14) and the property of the transformed demand matrix in (13); where (31c) follows by rearranging the oder of the summations; and where (31d) is because of (20) and since we set $\beta_{\overline{\mathcal{L}}} = -1$. Since (31) must hold for all $\{B_{\ell,\mathcal{T}} \in \mathbb{F}_{\mathsf{q}}^{\mathsf{B}/\binom{\mathsf{K}}{t}} : \ell \in \mathcal{L}, \ \mathcal{T} \in \Omega^{t}_{[\mathsf{K}]}\}$, and by the definition of transformed demand matrix in (12), we equivalently rewrite (31) as

$$\sum_{k\in\overline{\mathcal{T}}} \beta_{\{k\}\cup\mathcal{T}} \ \alpha_{k,\mathcal{T}} \ [\mathbb{D}']_{k,\ell} \tag{32a}$$

$$= \sum_{k\in\overline{\mathcal{T}}\cap\mathcal{L}} \beta_{\{k\}\cup\mathcal{T}} \ \alpha_{k,\mathcal{T}} \ 1_{\{k=\ell\}} + \sum_{k\in\overline{\mathcal{T}}\cap\overline{\mathcal{L}}} \beta_{\{k\}\cup\mathcal{T}} \ \alpha_{k,\mathcal{T}} \ x_{k,\ell} \tag{32b}$$

$$= 0 \in \mathbb{F}_{\mathsf{q}}, \quad \forall \ell \in \mathcal{L}, \ \forall \mathcal{T} \in \Omega^{t}_{[\mathsf{K}]}. \tag{32c}$$

We finally rewrite (32) by separating it into two cases

$$\sum_{k\in\overline{\mathcal{T}}\cap\overline{\mathcal{L}}} \beta_{\{k\}\cup\mathcal{T}} \ \alpha_{k,\mathcal{T}} \ x_{k,\ell} = \begin{cases} 0 & \forall \ell \in \mathcal{L}\cap\mathcal{T}, \ \forall \mathcal{T} \in \Omega^{t}_{[\mathsf{K}]} \\ -\beta_{\{\ell\}\cup\mathcal{T}} \ \alpha_{\ell,\mathcal{T}} & \forall \ell \in \mathcal{L}\cap\overline{\mathcal{T}}, \ \forall \mathcal{T} \in \Omega^{t}_{[\mathsf{K}]} \end{cases}. \tag{33}$$

Next, we say that a set $\mathcal{T} \subseteq [\mathsf{K}]$ is in "hierarchy $h$" if $|\mathcal{T}\cap\mathcal{L}| = h$ for some $h \in [0:\min(|\mathcal{T}|,|\mathcal{L}|)]$. We also say that $\beta_{\mathcal{S}}$ is in hierarchy $h$ if $\mathcal{S}$ is in hierarchy $h$. *We next seek to show that in general the decoding coefficients in hierarchy* $h+1$ *can be expressed as a linear combination of those in hierarchy* $h$.

*Initialization, or hierarchy* $h = 1$*:* $\beta_{\overline{\mathcal{L}}} = -1$ is the only decoding coefficient in hierarchy 0. Next we derive an expression for the decoding coefficients in hierarchy 1. By picking $\mathcal{T} = \overline{\mathcal{L}} \setminus \{u\}$, $u \in \overline{\mathcal{L}}$, and $\ell \in \mathcal{L}$ in (33) (and thus $\overline{\mathcal{T}} \cap \overline{\mathcal{L}} = \{u\}$), we express the decoding coefficients in hierarchy 1 as follows

$$\beta_{\{\ell\}\cup\overline{\mathcal{L}}\setminus\{u\}} = \frac{\alpha_{u,\overline{\mathcal{L}}\setminus\{u\}}}{\alpha_{\ell,\overline{\mathcal{L}}\setminus\{u\}}} \ x_{u,\ell}, \ \forall u \in \overline{\mathcal{L}}, \ \forall \ell \in \mathcal{L}. \tag{34}$$

*Hierarchy* $h$*:* From (33) with $\ell \in \mathcal{T}$, we have

$$\sum_{k\in\overline{\mathcal{T}}\cap\overline{\mathcal{L}}} \beta_{\{k\}\cup\mathcal{T}} \ \alpha_{k,\mathcal{T}} \ x_{k,\ell} = 0, \forall \ell \in \mathcal{L}\cap\mathcal{T}, \ \forall \mathcal{T} \in \Omega^{t}_{[\mathsf{K}]}. \tag{35}$$

In particular, for a fixed $\mathcal{T}$ in hierarchy $h > 0$, WLOG we let (recall that here $|\overline{\mathcal{L}}| = \mathsf{K} - r = t + 1 = |\mathcal{T}| + 1$ and thus $|\mathcal{T} \cap \mathcal{L}| = h$, $|\mathcal{T} \cap \overline{\mathcal{L}}| = t - h$, $|\overline{\mathcal{T}} \cap \mathcal{L}| = r - h$, $|\overline{\mathcal{T}} \cap \overline{\mathcal{L}}| = h + 1$)

$$\mathcal{T} \cap \mathcal{L} = \{\ell_1, \ldots, \ell_h\} : \ell_1 < \ldots < \ell_h, \tag{36a}$$
$$\overline{\mathcal{T}} \cap \overline{\mathcal{L}} = \{j_1, \ldots, j_h, j_{h+1}\} : j_1 < \ldots < j_{h+1}, \tag{36b}$$

and collect the $h$ constraints in (35) in matrix form as report in (37) at the top of the next page, which we equivalently re-write (by re-arranging some terms in the matrix equation) as in (38) at the top of the next page. By Cramer's rule, the solution of (38) can be written as

$$\begin{aligned} &(-1)^{h+1-i} \frac{\det\left(\mathbb{D}'[\overline{\mathcal{T}} \cap \overline{\mathcal{L}} \setminus \{j_i\}, \mathcal{L} \cap \mathcal{T}]\right)}{\det\left(\mathbb{D}'[\overline{\mathcal{T}} \cap \overline{\mathcal{L}} \setminus \{j_{h+1}\}, \mathcal{L} \cap \mathcal{T}]\right)} \\ &= \frac{\beta_{\{j_i\} \cup \mathcal{T}} \ \alpha_{j_i, \mathcal{T}}}{\beta_{\{j_{h+1}\} \cup \mathcal{T}} \ \alpha_{j_{h+1}, \mathcal{T}}}, \quad \forall i \in [h], \end{aligned} \tag{39}$$

or equivalently (39) can be written as (recall $j_i \in \overline{\mathcal{T}} \cap \overline{\mathcal{L}}$ for all $i \in [h+1]$)

$$(-1)^1 \frac{\beta_{\{j_1\} \cup \mathcal{T}} \ \alpha_{j_1, \mathcal{T}}}{\det\left(\mathbb{D}'[\overline{\mathcal{T}} \cap \overline{\mathcal{L}} \setminus \{j_1\}, \mathcal{L} \cap \mathcal{T}]\right)} \tag{40a}$$
$$= (-1)^2 \frac{\beta_{\{j_2\} \cup \mathcal{T}} \ \alpha_{j_2, \mathcal{T}}}{\det\left(\mathbb{D}'[\overline{\mathcal{T}} \cap \overline{\mathcal{L}} \setminus \{j_2\}, \mathcal{L} \cap \mathcal{T}]\right)} \tag{40b}$$
$$\vdots$$
$$= (-1)^{h+1} \frac{\beta_{\{j_{h+1}\} \cup \mathcal{T}} \ \alpha_{j_{h+1}, \mathcal{T}}}{\det\left(\mathbb{D}'[\overline{\mathcal{T}} \cap \overline{\mathcal{L}} \setminus \{j_{h+1}\}, \mathcal{L} \cap \mathcal{T}]\right)}, \tag{40c}$$

that is

$$\begin{aligned} &(-1)^{\mathsf{Ind}_{\overline{\mathcal{T}} \cap \overline{\mathcal{L}}, j}} \frac{\beta_{\{j\} \cup \mathcal{T}} \ \alpha_{j, \mathcal{T}}}{\det\left(\mathbb{D}'[\overline{\mathcal{T}} \cap \overline{\mathcal{L}} \setminus \{j\}, \mathcal{L} \cap \mathcal{T}]\right)} = \mathsf{c}_{\mathcal{T}}^{(\overline{\mathcal{L}})}, \\ &\qquad \forall j \in \overline{\mathcal{T}} \cap \overline{\mathcal{L}}, \forall \mathcal{T} \in \Omega^t_{[\mathsf{K}]}, \end{aligned} \tag{41}$$

for some constant $\mathsf{c}_{\mathcal{T}}^{(\overline{\mathcal{L}})} \in \mathbb{F}_{\mathsf{q}}$. All the decoding coefficients in (41) are in hierarchy $h$ if the set $\mathcal{T}$ is hierarchy $h$.

*Hierarchy $h+1$:* We plug the decoding coefficients in hierarchy $h$ from (41) into (33) with $\ell \in \overline{\mathcal{T}}$ and, by definition of determinant (i.e., Laplace expansion along a column), for a fixed $\mathcal{T} \in \Omega^t_{[\mathsf{K}]}$ we obtain (42), which is at the top of the next page, that is

$$\begin{aligned} &(-1)^{1+\mathsf{Ind}_{\mathcal{L} \cap \mathcal{T} \cup \{\ell\}, \ell}} \frac{\beta_{\{\ell\} \cup \mathcal{T}} \ \alpha_{\ell, \mathcal{T}}}{\det\left(\mathbb{D}'[\overline{\mathcal{T}} \cap \overline{\mathcal{L}}, \mathcal{L} \cap \mathcal{T} \cup \{\ell\}]\right)} = \mathsf{c}_{\mathcal{T}}^{(\overline{\mathcal{L}})}, \\ &\qquad \forall \ell \in \overline{\mathcal{T}} \cap \mathcal{L}, \forall \mathcal{T} \in \Omega^t_{[\mathsf{K}]}. \end{aligned} \tag{43}$$

where the constant $\mathsf{c}_{\mathcal{T}}^{(\overline{\mathcal{L}})}$ in (43) is the same as in (41). Notice that all the decoding coefficients in (43) are in hierarchy $h+1$ if the set $\mathcal{T}$ is hierarchy $h$.

*Combing everything together:* We can interpret (41) and (43) as follows: for a set $\mathcal{T} \in \Omega^t_{[\mathsf{K}]}$ and an element $k \in \overline{\mathcal{T}}$, we create a set $\mathcal{S} = \mathcal{T} \cup \{k\} \in \Omega^{t+1}_{[\mathsf{K}]}$ that satisfies the following:

- add a non-leader

$$\begin{aligned} &k = j \in \overline{\mathcal{T}} \cap \overline{\mathcal{L}} : \overline{\mathcal{T}} \cap \overline{\mathcal{L}} \setminus \{j\} = \overline{\mathcal{L}} \setminus (\{j\} \cup \mathcal{T}), \\ &\qquad \mathcal{L} \cap \mathcal{T} = (\{j\} \cup \mathcal{T}) \setminus \overline{\mathcal{L}}, \end{aligned} \tag{44a}$$

- add a leader

$$\begin{aligned} &k = \ell \in \overline{\mathcal{T}} \cap \mathcal{L} : \overline{\mathcal{T}} \cap \overline{\mathcal{L}} = \overline{\mathcal{L}} \setminus (\{\ell\} \cup \mathcal{T}), \\ &\qquad \mathcal{L} \cap \mathcal{T} \cup \{\ell\} = (\{\ell\} \cup \mathcal{T}) \setminus \overline{\mathcal{L}}. \end{aligned} \tag{44b}$$

Therefore (recall $\overline{\mathcal{T}} \cap \overline{\mathcal{L}} = \overline{\mathcal{L}} \setminus \mathcal{T}$, $\overline{\mathcal{T}} \cap \mathcal{L} = \mathcal{L} \setminus \mathcal{T}$) $\forall \mathcal{T} \in \Omega^t_{[\mathsf{K}]}$, $\forall k \in \overline{\mathcal{T}}$ we have (now we add again the superscript $\overline{\mathcal{L}}$ to the decoding coefficients)

$$\widetilde{\beta}^{(\overline{\mathcal{L}})}_{\{k\} \cup \mathcal{T}} := \frac{\beta^{(\overline{\mathcal{L}})}_{\{k\} \cup \mathcal{T}}}{\det\left(\mathbb{D}'[\overline{\mathcal{L}} \setminus (\{k\} \cup \mathcal{T}), (\{k\} \cup \mathcal{T}) \setminus \overline{\mathcal{L}}]\right)}, \tag{45a}$$
$$\phi^{(\overline{\mathcal{L}})}_{k, \mathcal{T}} := \begin{cases} \mathsf{Ind}_{\overline{\mathcal{L}} \setminus \mathcal{T}, k} & k \in \overline{\mathcal{L}} \setminus \mathcal{T} \\ 1 + \mathsf{Ind}_{(\{k\} \cup \mathcal{T}) \setminus \overline{\mathcal{L}}, k} & k \in \mathcal{L} \setminus \mathcal{T} \end{cases}, \tag{45b}$$
$$(-1)^{\phi^{(\overline{\mathcal{L}})}_{k, \mathcal{T}}} \cdot \alpha_{k, \mathcal{T}} \cdot \widetilde{\beta}^{(\overline{\mathcal{L}})}_{\{k\} \cup \mathcal{T}} = \mathsf{c}_{\mathcal{T}}^{(\overline{\mathcal{L}})}, \tag{45c}$$

for some constants $\{\mathsf{c}_{\mathcal{T}}^{(\overline{\mathcal{L}})} : \mathcal{T} \in \Omega^t_{[\mathsf{K}]}\}$. Note that the relationships in (45) are the same as those in (27) in Theorem 2 for the special case $\mathcal{A} = \overline{\mathcal{L}}$.

The term $\widetilde{\beta}^{(\overline{\mathcal{L}})}_{\{k\} \cup \mathcal{T}}$ in (45a) (that only depends on $\{k\} \cup \mathcal{T}$ as opposed to on both $k$ and $\mathcal{T}$) can be further expressed as a function of the encoding coefficients as follows. For a set $\mathcal{S} \in \Omega^{t+1}_{[\mathsf{K}]}, \mathcal{S} \neq \overline{\mathcal{L}}$, in hierarchy $h$ and by setting WLOG

$$\mathcal{S} \cap \mathcal{L} = \{\ell_1, \ldots, \ell_h\} : \ell_1 < \ldots < \ell_h, \tag{46a}$$
$$\overline{\mathcal{S}} \cap \overline{\mathcal{L}} = \{j_1, \ldots, j_h\} : j_1 < \ldots < j_h, \tag{46b}$$
$$\overline{\mathcal{L}} = \{j_1, \ldots, j_h\} \cup \mathcal{J}, \tag{46c}$$

i.e., $\mathcal{S} \cap \overline{\mathcal{L}} = \mathcal{J}$ and $\mathcal{S} = \{\ell_1 \ldots \ell_h\} \cup \mathcal{J}$, we iteratively use (43) to express the decoding coefficient with $\mathcal{S} = \{\ell_1 \ldots \ell_h\} \cup \mathcal{J}$ as (47), which is at the top of the next page. The last equality in (47) follows since by definition $\beta_{\{j_h \ldots j_1\} \cup \mathcal{J}} = \beta_{\overline{\mathcal{L}}} = -1$ and by convention $\det\left(\mathbb{D}'[\emptyset, \emptyset]\right) = 1$. *The relationship in* (47) *shows that each decoding coefficient is proportional to the determinant of a sub-matrix of the transformed demand matrix, and that the proportionality coefficient (denoted as $\widetilde{\beta}^{(\overline{\mathcal{L}})}_{\{\ell_1 \ldots \ell_h\} \cup \mathcal{J}}$) depends only on the encoding coefficients; the encoding coefficients however are not all free to vary, as they need to satisfy the relationships imposed by* (45c).

This concludes the proof for the case $\mathsf{K} - r = t + 1$.

### B. *Case* $\mathsf{K} - |\mathcal{L}| > t + 1$

It is easy to see that in order to locally reconstruct all non-sent multicast messages, we need not sum over all sent multicast messages indexed by $\{\mathcal{S} \in \Omega^{t+1}_{[\mathsf{K}]} : |\mathcal{S} \cap \mathcal{L}| > 0\}$ but only on those indexed by $\{\mathcal{S} \in \Omega^{t+1}_{\mathcal{A} \cup \mathcal{L}} : \mathcal{S} \neq \mathcal{A}\}$. By doing so, we can equivalently re-write (20) as

$$0 = \sum_{\mathcal{S} \in \Omega^{t+1}_{\mathcal{A} \cup \mathcal{L}}} \beta^{(\mathcal{A})}_{\mathcal{S}} W_{\mathcal{S}} : \ \beta^{(\mathcal{A})}_{\mathcal{A}} = -1, \ \forall \mathcal{A} \in \Omega^{t+1}_{[\mathsf{K}] \setminus \mathcal{L}}, \tag{48}$$

where in (48) the summation is over the $\binom{|\mathcal{L}|+t+1}{t+1}$ subsets of $\mathcal{A} \cup \mathcal{L}$ rather than over $\binom{\mathsf{K}}{t+1} - \binom{\mathsf{K}-|\mathcal{L}|}{t+1}$ terms in (20). In other words, for reconstructing non-sent multicast messages $W_{\mathcal{A}}$, we can consider a "reduced system" with users indexed by $\mathcal{A} \cup \mathcal{L}$ for which $W_{\mathcal{A}}$ is only non-sent multicast message

$$\left[\beta_{\{j_1\}\cup\mathcal{T}}\ \alpha_{j_1,\mathcal{T}} \quad \cdots \quad \beta_{\{j_h\}\cup\mathcal{T}}\ \alpha_{j_h,\mathcal{T}} \quad \beta_{\{j_{h+1}\}\cup\mathcal{T}}\ \alpha_{j_{h+1},\mathcal{T}}\right] \underbrace{\begin{bmatrix} x_{j_1,\ell_1} & \cdots & x_{j_1,\ell_h} \\ \vdots & \ddots & \vdots \\ x_{j_h,\ell_1} & \cdots & x_{j_h,\ell_h} \\ x_{j_{h+1},\ell_1} & \cdots & x_{j_{h+1},\ell_h} \end{bmatrix}}_{=\mathbb{D}'[\overline{\mathcal{T}}\cap\overline{\mathcal{L}},\mathcal{L}\cap\mathcal{T}]\in\mathbb{F}_{\mathsf{q}}^{h+1\times h}} = 0 \in \mathbb{F}_{\mathsf{q}}^{1\times h}, \tag{37}$$

$$\left[\frac{\beta_{\{j_1\}\cup\mathcal{T}}\ \alpha_{j_1,\mathcal{T}}}{\beta_{\{j_{h+1}\}\cup\mathcal{T}}\ \alpha_{j_{h+1},\mathcal{T}}} \quad \cdots \quad \frac{\beta_{\{j_h\}\cup\mathcal{T}}\ \alpha_{j_h,\mathcal{T}}}{\beta_{\{j_{h+1}\}\cup\mathcal{T}}\ \alpha_{j_{h+1},\mathcal{T}}}\right] \underbrace{\begin{bmatrix} x_{j_1,\ell_1} & \cdots & x_{j_1,\ell_h} \\ \vdots & \ddots & \vdots \\ x_{j_h,\ell_1} & \cdots & x_{j_h,\ell_h} \end{bmatrix}}_{=\mathbb{D}'[\overline{\mathcal{T}}\cap\overline{\mathcal{L}}\setminus\{j_{h+1}\},\mathcal{L}\cap\mathcal{T}]\in\mathbb{F}_{\mathsf{q}}^{h\times h}} = -\underbrace{\left[x_{j_{h+1},\ell_1} \quad \ldots \quad x_{j_{h+1},\ell_h}\right]}_{=\mathbb{D}'[\{j_{h+1}\},\mathcal{L}\cap\mathcal{T}]\in\mathbb{F}_{\mathsf{q}}^{1\times h}}, \tag{38}$$

$$\begin{aligned} -\beta_{\{\ell\}\cup\mathcal{T}}\ \alpha_{\ell,\mathcal{T}} &= \sum_{k\in\overline{\mathcal{T}}\cap\overline{\mathcal{L}}} \beta_{\{k\}\cup\mathcal{T}}\ \alpha_{k,\mathcal{T}}\ x_{k,\ell} && \text{(42a)} \\ &= \frac{\beta_{\{j_{h+1}\}\cup\mathcal{T}}\ \alpha_{j_{h+1},\mathcal{T}}}{\det\left(\mathbb{D}'[\overline{\mathcal{T}}\cap\overline{\mathcal{L}}\setminus\{j_{h+1}\},\mathcal{L}\cap\mathcal{T}]\right)} \cdot \sum_{i\in[h+1]} (-1)^{h+1-i} \det\left(\mathbb{D}'[\overline{\mathcal{T}}\cap\overline{\mathcal{L}}\setminus\{j_i\},\mathcal{L}\cap\mathcal{T}]\right)\ x_{j_i,\ell} && \text{(42b)} \\ &= (-1)^{h+1} \frac{\beta_{\{j_{h+1}\}\cup\mathcal{T}}\ \alpha_{j_{h+1},\mathcal{T}}}{\det\left(\mathbb{D}'[\overline{\mathcal{T}}\cap\overline{\mathcal{L}}\setminus\{j_{h+1}\},\mathcal{L}\cap\mathcal{T}]\right)} \cdot (-1)^{-\mathsf{Ind}_{\mathcal{L}\cap\mathcal{T}\cup\{\ell\},\ell}} \det\left(\mathbb{D}'[\overline{\mathcal{T}}\cap\overline{\mathcal{L}},\mathcal{L}\cap\mathcal{T}\cup\{\ell\}]\right) && \text{(42c)} \\ &= \mathsf{c}_{\mathcal{T}}^{(\overline{\mathcal{L}})} \cdot (-1)^{-\mathsf{Ind}_{\mathcal{L}\cap\mathcal{T}\cup\{\ell\},\ell}}\ \det\left(\mathbb{D}'[\overline{\mathcal{T}}\cap\overline{\mathcal{L}},\mathcal{L}\cap\mathcal{T}\cup\{\ell\}]\right),\ \forall\ell\in\overline{\mathcal{T}}\cap\mathcal{L},\forall\mathcal{T}\in\Omega_{[\mathsf{K}]}^{t}, && \text{(42d)} \end{aligned}$$

$$\begin{aligned} \widetilde{\beta}_{\mathcal{S}}^{(\overline{\mathcal{L}})} = \widetilde{\beta}_{\{\ell_1\ldots\ell_h\}\cup\mathcal{J}}^{(\overline{\mathcal{L}})} &= \frac{\beta_{\{\ell_1\ldots\ell_h\}\cup\mathcal{J}}}{\det\left(\mathbb{D}'[\overline{\mathcal{S}}\cap\overline{\mathcal{L}},\mathcal{S}\cap\mathcal{L}]\right)} = -\frac{\alpha_{j_h,\{\ell_1\ldots\ell_{h-1}\}\cup\mathcal{J}}}{\alpha_{\ell_h,\{\ell_1\ldots\ell_{h-1}\}\cup\mathcal{J}}}\ \frac{\beta_{\{j_h\}\cup\{\ell_1\ldots\ell_{h-1}\}\cup\mathcal{J}}}{\det\left(\mathbb{D}'[\overline{\mathcal{S}}\cap\overline{\mathcal{L}}\setminus\{j_h\},\mathcal{S}\cap\mathcal{L}\setminus\{\ell_h\}]\right)} && \text{(47a)} \\ &= (-1)^h\ \frac{\alpha_{j_h,\{\ell_1\ldots\ell_{h-1}\}\cup\mathcal{J}}}{\alpha_{\ell_h,\{\ell_1\ldots\ell_{h-1}\}\cup\mathcal{J}}}\ \frac{\alpha_{j_{h-1},\{j_h\}\cup\{\ell_1\ldots\ell_{h-2}\}\cup\mathcal{J}}}{\alpha_{\ell_{h-1},\{j_h\}\cup\{\ell_1\ldots\ell_{h-2}\}\cup\mathcal{J}}} \cdots \frac{\alpha_{j_1,\{j_h\ldots j_2\}\cup\mathcal{J}}}{\alpha_{\ell_1,\{j_h\ldots j_2\}\cup\mathcal{J}}}\ \frac{\beta_{\{j_h\ldots j_1\}\cup\mathcal{J}}}{\det\left(\mathbb{D}'[\emptyset,\emptyset]\right)} && \text{(47b)} \\ &= (-1)^{h+1}\ \prod_{i=1}^{h} \frac{\alpha_{j_i,\{j_h\ldots j_{i+1}\}\cup\{\ell_1\ldots\ell_{i-1}\}\cup\mathcal{J}}}{\alpha_{\ell_i,\{j_h\ldots j_{i+1}\}\cup\{\ell_1\ldots\ell_{i-1}\}\cup\mathcal{J}}}, && \text{(47c)} \end{aligned}$$

to be reconstructed by the non-leader users indexed by $\mathcal{A}$. The analysis we did in Section IV-A applies to this "reduced system" with $|\mathcal{A}\cup\mathcal{L}| = t+1+r$ users. After substituting $\mathcal{A}$ instead of $\overline{\mathcal{L}}$, the conditions in (45) are as stated in in (27) in Theorem 2. This concluded the proof.

## V. Proof of Propositions 1 and 2

In order to clarify our proposed approach, which is aimed to identify how to choose the encoding coefficients that are free to vary and which one are therefore determined by the relationships identified by (27) in Theorem 2, we continue with our example in Section III-B.

### A. *Example of SLFR system with parameters* $(\mathsf{N},\mathsf{K},|\mathcal{L}|,t) = (2,4,2,1)$*: algebraic manipulations*

We continue our example of the SLFR problem with $\mathsf{N}=2$ files, $\mathsf{K}=4$ users, $|\mathcal{L}|=2$ leader users, and memory size $t=1$. The steps reported next are as in the proof of Theorem 2 in the previous Section.

The multicast messages sent by the server that are useful for the leader users indexed by $\mathcal{L}=\{1,2\}$ are those in (28), except $W_{\{3,4\}}$. The multicast message $W_{\{3,4\}}$ is not sent because it is only useful for the non-leader users. In order to reconstruct $W_{\{3,4\}}$ at the non-leader users, we seek to find decoding coefficients $\{\beta_{\mathcal{S}}^{\{3,4\}} : \mathcal{S}\in\Omega_{[4]}^{2}, \mathcal{S}\neq\{3,4\}\}$ such that $W_{\{3,4\}}$ can be written as

$$\begin{aligned} W_{\{3,4\}} = \beta_{\{1,2\}}^{\{3,4\}} W_{\{1,2\}} + \beta_{\{1,3\}}^{\{3,4\}} W_{\{1,3\}} + \beta_{\{1,4\}}^{\{3,4\}} W_{\{1,4\}} \\ + \beta_{\{2,3\}}^{\{3,4\}} W_{\{2,3\}} + \beta_{\{2,4\}}^{\{3,4\}} W_{\{2,4\}}. \end{aligned} \tag{49}$$

In other words, we aim to solve the equation in (50) at the top of the next page (where we use the definition of multicast messages and transformed demand matrix) for any realization of the demand-blocks, and where we set $\beta_{\{3,4\}}^{\{3,4\}} = -1$ (which is only decoding coefficient in hierarchy 0 corresponding to $\mathcal{S}=\{3,4\}$). This can be done by equating the scalar coefficient that multiplies each demand-block in (50) to zero as follows.

We start with the decoding coefficients in hierarchy 1, namely those decoding coefficients $\beta_{\mathcal{S}}^{\{3,4\}}$'s for $\mathcal{S}\in\{\{1,3\},\{1,4\},\{2,3\},\{2,4\}\}$ (since each such set $\mathcal{S}$ satisfies $|\mathcal{S}\cap\mathcal{L}| = |\mathcal{S}\cap\{1,2\}| = 1$). These $\mathcal{S}$'s can be obtained in two

$$
\begin{aligned}
0 &= \beta^{\{3,4\}}_{\{1,2\}}\left(\alpha_{1,\{2\}}B_{1,\{2\}} + \alpha_{2,\{1\}}B_{2,\{1\}}\right) && \text{(50a)}\\
&+ \beta^{\{3,4\}}_{\{3,4\}}\left(\alpha_{3,\{4\}}[x_{3,1}B_{1,\{4\}} + x_{3,2}B_{2,\{4\}}] + \alpha_{4,\{3\}}[x_{4,1}B_{1,\{3\}} + x_{4,2}B_{2,\{3\}}]\right) && \text{(50b)}\\
&+ \beta^{\{3,4\}}_{\{1,3\}}\left(\alpha_{1,\{3\}}B_{1,\{3\}} + \alpha_{3,\{1\}}[x_{3,1}B_{1,\{1\}} + x_{3,2}B_{2,\{1\}}]\right) + \beta^{\{3,4\}}_{\{1,4\}}\left(\alpha_{1,\{4\}}B_{1,\{4\}} + \alpha_{4,\{1\}}[x_{4,1}B_{1,\{1\}} + x_{4,2}B_{2,\{1\}}]\right) && \text{(50c)}\\
&+ \beta^{\{3,4\}}_{\{2,3\}}\left(\alpha_{2,\{3\}}B_{2,\{3\}} + \alpha_{3,\{2\}}[x_{3,1}B_{1,\{2\}} + x_{3,2}B_{2,\{2\}}]\right) + \beta^{\{3,4\}}_{\{2,4\}}\left(\alpha_{2,\{4\}}B_{2,\{4\}} + \alpha_{4,\{2\}}[x_{4,1}B_{1,\{2\}} + x_{4,2}B_{2,\{2\}}]\right) && \text{(50d)}
\end{aligned}
$$

ways: (Case a) by adding a leader user to either $\mathcal{T} = \{3\}$ or $\mathcal{T} = \{4\}$, or (Case b) by adding a non-leader user to either $\mathcal{T} = \{1\}$ or $\mathcal{T} = \{2\}$. We look at these two cases separately.

- **Case a):** Decoding Coefficients in Hierarchy 1 from a $\mathcal{T} \in \Omega^1_{[4]}$ in Hierarchy 0. By focusing on some demand-blocks in (50), we equate their scalar coefficient to zero to get

$$
\begin{aligned}
&\text{for } B_{1,\{3\}}: \ \alpha_{4,\{3\}}x_{4,1} = \beta^{\{3,4\}}_{\{1,3\}}\alpha_{1,\{3\}} && \text{(51a)}\\
&\iff \frac{\beta^{\{3,4\}}_{\{1,3\}}}{x_{4,1}} = \frac{\alpha_{4,\{3\}}}{\alpha_{1,\{3\}}} = \widetilde{\beta}^{\{3,4\}}_{\{1,3\}}, && \text{(51b)}\\
&\text{for } B_{1,\{4\}}: \ \alpha_{3,\{4\}}x_{3,1} = \beta^{\{3,4\}}_{\{1,4\}}\alpha_{1,\{4\}} && \text{(51c)}\\
&\iff \frac{\beta^{\{3,4\}}_{\{1,4\}}}{x_{3,1}} = \frac{\alpha_{3,\{4\}}}{\alpha_{1,\{4\}}} = \widetilde{\beta}^{\{3,4\}}_{\{1,4\}}, && \text{(51d)}\\
&\text{for } B_{2,\{3\}}: \ \alpha_{4,\{3\}}x_{4,2} = \beta^{\{3,4\}}_{\{2,3\}}\alpha_{2,\{3\}} && \text{(51e)}\\
&\iff \frac{\beta^{\{3,4\}}_{\{2,3\}}}{x_{4,2}} = \frac{\alpha_{4,\{3\}}}{\alpha_{2,\{3\}}} = \widetilde{\beta}^{\{3,4\}}_{\{2,3\}}, && \text{(51f)}\\
&\text{for } B_{2,\{4\}}: \ \alpha_{3,\{4\}}x_{3,2} = \beta^{\{3,4\}}_{\{2,4\}}\alpha_{2,\{4\}} && \text{(51g)}\\
&\iff \frac{\beta^{\{3,4\}}_{\{2,4\}}}{x_{3,2}} = \frac{\alpha_{3,\{4\}}}{\alpha_{2,\{4\}}} = \widetilde{\beta}^{\{3,4\}}_{\{2,4\}}, && \text{(51h)}
\end{aligned}
$$

where (51b) is equivalent to (34) for $u = 4, \ell = 1$, where (51d) is equivalent to (34) for $u = 3, \ell = 1$, where (51f) is equivalent to (34) for $u = 4, \ell = 2$, and where (51d) is equivalent to (34) for $u = 3, \ell = 2$.

The relationships in (51) correspond to the initialization step in (34); they can also be seen as an iteration in (42) with $\mathsf{c}^{\{3,4\}}_{\{3\}} = \alpha_{4,\{3\}}$ and $\mathsf{c}^{\{3,4\}}_{\{4\}} = \alpha_{3,\{4\}}$.

- **Case b):** Decoding Coefficients in Hierarchy 1 from a $\mathcal{T} \in \Omega^1_{[4]}$ in Hierarchy 1. The decoding coefficients we just derived in (51) also appear in other terms in (50). In particular

$$
\begin{aligned}
&\text{for } B_{1,\{1\}}: \quad 0 = \beta^{\{3,4\}}_{\{1,3\}}\alpha_{3,\{1\}}x_{3,1} + \beta^{\{3,4\}}_{\{1,4\}}\alpha_{4,\{1\}}x_{4,1} && \text{(52a)}\\
&\iff -\frac{\beta^{\{3,4\}}_{\{1,3\}}\alpha_{3,\{1\}}}{x_{4,1}} = \frac{\beta^{\{3,4\}}_{\{1,4\}}\alpha_{4,\{1\}}}{x_{3,1}} = \mathsf{c}^{\{3,4\}}_{\{1\}} && \text{(52b)}\\
&\iff -\frac{\alpha_{4,\{3\}}\alpha_{3,\{1\}}}{\alpha_{1,\{3\}}} = \frac{\alpha_{3,\{4\}}\alpha_{4,\{1\}}}{\alpha_{1,\{4\}}} = \mathsf{c}^{\{3,4\}}_{\{1\}}, && \text{(52c)}\\
&\text{for } B_{2,\{2\}}: \quad 0 = \beta^{\{3,4\}}_{\{2,3\}}\alpha_{3,\{2\}}x_{3,2} + \beta^{\{3,4\}}_{\{2,4\}}\alpha_{4,\{2\}}x_{4,2} && \text{(52d)}\\
&\iff -\frac{\beta^{\{3,4\}}_{\{2,3\}}\alpha_{3,\{2\}}}{x_{4,2}} = \frac{\beta^{\{3,4\}}_{\{2,4\}}\alpha_{4,\{2\}}}{x_{3,2}} = \mathsf{c}^{\{3,4\}}_{\{2\}} && \text{(52e)}\\
&\iff -\frac{\alpha_{4,\{3\}}\alpha_{3,\{2\}}}{\alpha_{2,\{3\}}} = \frac{\alpha_{3,\{4\}}\alpha_{4,\{2\}}}{\alpha_{2,\{4\}}} = \mathsf{c}^{\{3,4\}}_{\{2\}}, && \text{(52f)}
\end{aligned}
$$

where (52a) is equivalent to (35) for $\mathcal{T} = \{1\}, \ell = 1$, where (52b) is equivalent to (40) for $\mathcal{T} = \{1\}, j_1 = 3, j_2 = 4$, where (52c) is by substituting (51b) and (51d), where (52d) is equivalent to (35) for $\mathcal{T} = \{2\}, \ell = 2$, where (52e) is equivalent to (40) for $\mathcal{T} = \{2\}, j_1 = 3, j_2 = 4$,where (52f) is by substituting (51f) and (51h).

We see that (52c) and (52f) are constraints that the encoding coefficients must satisfy in order to be able to locally reconstruct the non-sent multicast message.

We now look at the decoding coefficients in hierarchy 2. There is only one decoding coefficient in hierarchy 2, namely $\beta^{\{3,4\}}_{\mathcal{S}}$ for $\mathcal{S} = \{1, 2\}$. Set $\mathcal{S} = \{1, 2\}$ can be obtained in two ways: (Case a') by adding leader user $\ell = 2$ to $\mathcal{T} = \{1\}$, or (Case b') by adding leader user $\ell = 1$ to $\mathcal{T} = \{2\}$. We shall now look at these two cases separately.

- **Case a'):** we get

$$
\begin{aligned}
&\text{for } B_{2,\{1\}}: \ -\beta^{\{3,4\}}_{\{1,2\}}\alpha_{2,\{1\}} && \text{(53a)}\\
&= \beta^{\{3,4\}}_{\{1,3\}}\alpha_{3,\{1\}}x_{3,2} + \beta^{\{3,4\}}_{\{1,4\}}\alpha_{4,\{1\}}x_{4,2} && \text{(53b)}\\
&= \underbrace{(-x_{4,1}x_{3,2} + x_{3,1}x_{4,2})}_{=\det(\mathbb{D}'[\{3,4\},\{1,2\}])} \cdot \underbrace{\frac{\alpha_{3,\{4\}}\alpha_{4,\{1\}}}{\alpha_{1,\{4\}}}}_{=\mathsf{c}^{\{3,4\}}_{\{1\}}} && \text{(53c)}\\
&\iff \frac{\beta^{\{3,4\}}_{\{1,2\}}}{x_{3,1}x_{4,2} - x_{4,1}x_{3,2}} = -\frac{\alpha_{3,\{4\}}\alpha_{4,\{1\}}}{\alpha_{1,\{4\}}\alpha_{2,\{1\}}}\\
&\qquad = -\frac{\mathsf{c}^{\{3,4\}}_{\{1\}}}{\alpha_{2,\{1\}}} = \widetilde{\beta}^{\{3,4\}}_{\{1,2\}}, && \text{(53d)}
\end{aligned}
$$

where (53b) is equivalent to (42) for $\ell = 2, \mathcal{T} = \{1\}$, where (53c) is by substituting the values in (51b) and (51d) and by using the equality in (52c), and where (53d) is by rearranging the terms and is equivalent to (43) with $\ell = 2$ to $\mathcal{T} = \{1\}$.

- **Case b'):** as in (53) but by swapping the role of the leader users, we get

$$
\begin{aligned}
&\text{for } B_{1,\{2\}}: \ \frac{\beta^{\{3,4\}}_{\{1,2\}}}{x_{3,1}x_{4,2} - x_{4,1}x_{3,2}} = -\frac{\alpha_{4,\{3\}}\alpha_{3,\{2\}}}{\alpha_{2,\{3\}}\alpha_{1,\{2\}}} && \text{(54a)}\\
&\qquad = \frac{\mathsf{c}^{\{3,4\}}_{\{2\}}}{\alpha_{1,\{2\}}} = \widetilde{\beta}^{\{3,4\}}_{\{1,2\}}, && \text{(54b)}
\end{aligned}
$$

where (54b) is equivalent to (42) for $\ell = 1$ to $\mathcal{T} = \{2\}$.

We see that (53d) and (54b) give two different values for $\widetilde{\beta}^{\{3,4\}}_{\{1,2\}}$, therefore they impose a constraint on the encoding coefficients, which together with (52c) and (52f) result in

$$
\begin{aligned}
-\frac{\alpha_{4,\{3\}}\alpha_{3,\{1\}}}{\alpha_{1,\{3\}}\alpha_{2,\{1\}}} &\overset{\text{by (52c)}}{=} \frac{\alpha_{3,\{4\}}\alpha_{4,\{1\}}}{\alpha_{1,\{4\}}\alpha_{2,\{1\}}} && \text{(55a)}\\
&\overset{\text{by (53d) and (54b)}}{=} \frac{\alpha_{4,\{3\}}\alpha_{3,\{2\}}}{\alpha_{2,\{3\}}\alpha_{1,\{2\}}} && \text{(55b)}\\
&\overset{\text{by (52f)}}{=} -\frac{\alpha_{3,\{4\}}\alpha_{4,\{2\}}}{\alpha_{2,\{4\}}\alpha_{1,\{2\}}}. && \text{(55c)}
\end{aligned}
$$

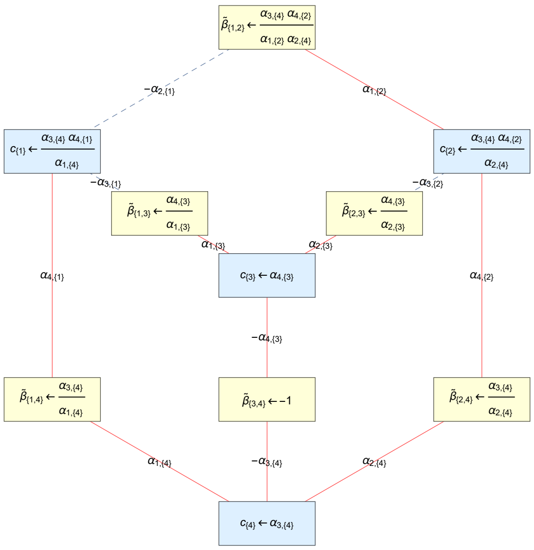

$\tilde\beta_{\{1,2\}} \leftarrow \frac{\alpha_{3,\{4\}}\,\alpha_{4,\{2\}}}{\alpha_{1,\{2\}}\,\alpha_{2,\{4\}}}$
$-\alpha_{2,\{1\}}$
$\alpha_{1,\{2\}}$
$c_{\{1\}} \leftarrow \frac{\alpha_{3,\{4\}}\,\alpha_{4,\{1\}}}{\alpha_{1,\{4\}}}$
$c_{\{2\}} \leftarrow \frac{\alpha_{3,\{4\}}\,\alpha_{4,\{2\}}}{\alpha_{2,\{4\}}}$
$-\alpha_{3,\{1\}}$
$-\alpha_{3,\{2\}}$
$\tilde\beta_{\{1,3\}} \leftarrow \frac{\alpha_{4,\{3\}}}{\alpha_{1,\{3\}}}$
$\tilde\beta_{\{2,3\}} \leftarrow \frac{\alpha_{4,\{3\}}}{\alpha_{2,\{3\}}}$
$\alpha_{1,\{3\}}$
$\alpha_{2,\{3\}}$
$\alpha_{4,\{1\}}$
$c_{\{3\}} \leftarrow \alpha_{4,\{3\}}$
$\alpha_{4,\{2\}}$
$-\alpha_{4,\{3\}}$
$\tilde\beta_{\{1,4\}} \leftarrow \frac{\alpha_{3,\{4\}}}{\alpha_{1,\{4\}}}$
$\tilde\beta_{\{3,4\}} \leftarrow -1$
$\tilde\beta_{\{2,4\}} \leftarrow \frac{\alpha_{3,\{4\}}}{\alpha_{2,\{4\}}}$
$\alpha_{1,\{4\}}$
$-\alpha_{3,\{4\}}$
$\alpha_{2,\{4\}}$
$c_{\{4\}} \leftarrow \alpha_{3,\{4\}}$

Fig. 3: A graph representing the relationships imposed by (45) in for the case $\mathsf{K} = 4, |\mathcal{L}| = 2, t = 1$, with $\mathcal{L} = \{1, 2\}$ and $\mathcal{A} = \{3, 4\}$. The superscript $\mathcal{A} = \{3, 4\}$ in the vertices is omitted in order not to clutter the figure.

To conclude, in order to be able to locally reconstruct the non-sent multicast message from the sent multicast messages, by rearranging the terms in (55), we find that 3 (out of 12) encoding coefficients must have a specific expression, for example

$$\text{by (52c) for } \mathsf{c}_1^{\{3,4\}} : \alpha_{3,\{1\}} = -\alpha_{1,\{3\}} \frac{\alpha_{4,\{1\}}}{\alpha_{1,\{4\}}} \frac{\alpha_{3,\{4\}}}{\alpha_{4,\{3\}}}, \tag{56a}$$

$$\text{by (52f) for } \mathsf{c}_2^{\{3,4\}} : \alpha_{3,\{2\}} = -\alpha_{2,\{3\}} \frac{\alpha_{4,\{2\}}}{\alpha_{2,\{4\}}} \frac{\alpha_{3,\{4\}}}{\alpha_{4,\{3\}}}, \tag{56b}$$

$$\text{by (55) for } \widetilde{\beta}_{\{1,2\}}^{\{3,4\}} : \alpha_{2,\{1\}} = -\frac{\alpha_{4,\{1\}}}{\alpha_{1,\{4\}}} \frac{\alpha_{2,\{4\}}}{\alpha_{4,\{2\}}} \alpha_{1,\{2\}}, \tag{56c}$$

which means that only 9 encoding coefficients are free to vary (they also must be non-zero) and the remaining 3 must be a function of those 9. *The question we ask in this section is whether the encoding coefficients that are free to vary can be found in an efficient way, as opposed to 'by inspection' as we have done in this simple example. The answer is in the positive and the key is to visualize the constraints imposed by Theorem 2 on a graph, as we shall explain next.*

### *B. Example of SLFR system with parameters* $(\mathsf{N}, \mathsf{K}, |\mathcal{L}|, t) = (2, 4, 2, 1)$*: graph representation*

Fig. 3 shows a graph representing the relationships imposed by (45) for the case of $\mathsf{K} = 4$ users, $|\mathcal{L}| = 2$ leader users, and memory size $t = 1$, where WLOG $\mathcal{L} = \{1, 2\}$ and thus $\mathcal{A} = \{3, 4\}$. For legibility, we omitted the superscript $\mathcal{A} = \{3, 4\}$ from the vertices in Fig. 3 and in the notation in the rest of this subsection. The vertices of the graph are $\{\widetilde{\beta}_{\mathcal{S}} : \mathcal{S} \in \Omega^2_{[4]}\}$, which are the 6 yellow boxes in Fig. 3, and $\{\mathsf{c}_{\mathcal{T}} : \mathcal{T} \in \Omega^1_{[4]}\}$, which are the 4 cyan boxes in Fig. 3. The graph is undirected and bipartite and does not depend on the demand matrix. The edges are labeled by an encoding coefficient with an appropriate sign according to (45c).

We note that the so obtained graph has cycles, where the encoding coefficients on the edges on a cycle are constrained by (45). Therefore, a spanning tree[2] for the graph identifies the encoding coefficients that are free to vary (those that label the edges of the spanning tree) while the remaining ones must be of a specific form adhering to (45). For the example in Fig. 3, the solid red edges form a spanning tree; the expression on the RHS of the symbol ← in a vertex box is the value of the vertex when we traverse the graph from the decoding coefficient in hierarchy 0 (here $\widetilde{\beta}_{\{3,4\}} = -1$, which we choose to be the root of a spanning tree) along the spanning tree. For the graph in Fig. 3, the 3 edges that are not in the spanning tree (marked by doted blue lines) correspond to the constraints in (56). This can be seen as follows.

A vertex on a cycle can "take" two different values depending from which side we reach said vertex and these two values must be equal; therefore each cycle is a constraint. In Fig. 3 by starting from vertex $\widetilde{\beta}_{\{3,4\}} = -1$, we arrive at the hierarchy 1 coefficients in $\{\widetilde{\beta}_{\{\ell,j\}}, \ell \in \mathcal{L} = \{1, 2\}, j \in \mathcal{A} = \{3, 4\}\}$ through $\mathsf{c}_{\{3\}} = \alpha_{4,\{3\}}$ and $\mathsf{c}_{\{4\}} = \alpha_{3,\{4\}}$ (see also comment right after eq.(51)). At this point we see there are two cycles: starting from the vertex $\widetilde{\beta}_{\{3,4\}}$ along the edge with label $-\alpha_{4,\{3\}}$, one is clockwise (towards vertex $\widetilde{\beta}_{\{2,3\}}$) and the other is counterclockwise (towards vertex $\widetilde{\beta}_{\{1,3\}}$). In order to get a spanning tree, we do not include the edge with label $-\alpha_{3,\{1\}}$ (for the counterclockwise cycle) and the edge with label $-\alpha_{3,\{2\}}$ (for the clockwise cycle). We note that by imposing that the value of the vertex $\mathsf{c}_{\{1\}}$ (resp. $\mathsf{c}_{\{2\}}$) is the same even if we reach it through the left-out-edge $-\alpha_{3,\{1\}}$ (resp. $-\alpha_{3,\{2\}}$) we obtain the that $\alpha_{3,\{1\}}$ must take the value in (56a) (resp. $\alpha_{3,\{2\}}$ must take the value in (56b)). At this point the only vertex that has not been covered by the spanning tree is $\widetilde{\beta}_{\{1,2\}}$; we have two paths that lead to $\widetilde{\beta}_{\{1,2\}}$: (i) by proceeding from $\mathsf{c}_{\{2\}}$ along the edge with label $+\alpha_{1,\{2\}}$: this imposes the relationship in (54); while (ii) by proceeding from $\mathsf{c}_{\{1\}}$ along the edge with label $-\alpha_{2,\{1\}}$: this imposes the relationship in (53); but the two must be equal, thus we get that, for example, $\alpha_{1,\{2\}}$ must take the value in (56c).

We next generalize the ideas through a simple example.

### *C. Graph Representation – Proof of Proposition 1*

In Theorem 2, for a fix set $\mathcal{L}$ of leader users and for each non-sent multicast message indexed by the set $\mathcal{A} \in \Omega^{t+1}_{[\mathsf{K}]\setminus\mathcal{L}}$ of non-leader users, the relationships among the parameters in

$$\mathcal{V}_1^{(\mathcal{A})} := \{\mathsf{c}_{\mathcal{T}}^{(\mathcal{A})} : \mathcal{T} \in \Omega^t_{\mathcal{A}\cup\mathcal{L}}\}, \tag{57a}$$

$$\mathcal{V}_2^{(\mathcal{A})} := \{\widetilde{\beta}_{\mathcal{S}}^{(\mathcal{A})} : \mathcal{S} \in \Omega^{t+1}_{\mathcal{A}\cup\mathcal{L}}\}, \tag{57b}$$

imposed by (27) can be represented by an undirected bipartite graph $\mathcal{G}^{(\mathcal{A})}$, where $\mathcal{V}^{(\mathcal{A})} := \mathcal{V}_1^{(\mathcal{A})} \cup \mathcal{V}_2^{(\mathcal{A})}$ as defined in (57) is the vertex set and $\mathcal{E}^{(\mathcal{A})}$ is the edge set given by

$$\mathcal{E}^{(\mathcal{A})} := \{(\widetilde{\beta}^{(\mathcal{A})}_{\{k\}\cup\mathcal{T}}, \mathsf{c}^{(\mathcal{A})}_{\mathcal{T}}) :$$

[2] A spanning tree is a subset of the graph, which has *all the vertices of the graph covered with minimum possible number of edges*. Hence, a spanning tree does not have cycles and it cannot be disconnected. Moreover, every connected and undirected graph has at least one spanning tree.

$$\mathcal{T} \in \Omega^t_{\mathcal{A}\cup\mathcal{L}},\ k \in (\mathcal{A} \cup \mathcal{L}) \setminus \mathcal{T}\}. \quad (58)$$

We then label each edge with (up to a sign) an encoding coefficient to capture the relationships in (27), in particular we assign

$$\text{label} \quad (-1)^{\phi^{(\mathcal{A})}_{k,\mathcal{T}}}\alpha_{k,\mathcal{T}} \quad \text{to edge} \quad (\widetilde{\beta}^{(\mathcal{A})}_{\{k\}\cup\mathcal{T}}, \mathsf{c}^{(\mathcal{A})}_{\mathcal{T}}). \quad (59)$$

The labels in (59) are not to be interpreted as weights for the edges. The graph $\mathcal{G}^{(\mathcal{A})}$ is what we referred to earlier as graph-$\mathcal{A}$.

As we did in Section IV, also here we distinguish the case when there is only one non-sent multicast message to locally reconstruct from the case of strictly more than one.

*1) Case* $\mathsf{K} - |\mathcal{L}| = t+1$*:* We create a spanning tree to find values for all the vertices by using (45). We elect $\widetilde{\beta}^{(\mathcal{A})}_{\mathcal{A}}$ (recall here we have $\mathcal{A} = \overline{\mathcal{L}}$) to be the root node and assign to it the value $-1$. We then create a spanning tree from that root. By the properties of spanning trees, the encoding coefficients on the edges of the spanning tree are free to vary (i.e., they can be any non-zero value), while the encoding coefficients on edges that are not part of the spanning tree are determined through the following relationship: every path from the root to a node determines the value of the node by using (45) and all those values must be equal; in other words, every cycle in the graph, obtained by adding an edge that is not on the spanning tree to the spanning tree, is a constraint.

*2) Case* $\mathsf{K} - |\mathcal{L}| > t+1$*:* In the case $\mathsf{K} - |\mathcal{L}| = t + 1$, there is only one graph and finding a spanning tree on the graph identifies all the encoding coefficients that are free to vary. The situation is more complex when $\mathsf{K} - |\mathcal{L}| > t + 1$ as now there are multiple non-sent multicast messages; in this case, there are multiple disjoint components to the overall graph (one per each non-sent multicast message) that share the same encoding coefficients as edge labels. This more complex scenario is analyzed next as illustrated in the next example.

Fig. 4 shows the overall graph (composed of disconnected component $\mathcal{G}^{(\{3,4\})}$ and $\mathcal{G}^{(\{3,5\})}$ and $\mathcal{G}^{(\{4,5\})}$) and a set of possible spanning trees for the case $\mathsf{K} = 5, r = 2, t = 1$, by using the same convention as in Fig. 3. Unlike the case $\mathsf{K} - |\mathcal{L}| = t + 1$, here some encoding coefficients appear more than once in the overall graph, meaning that finding a spanning tree independently for each disconnect component may result in some encoding coefficients being part of one spanning tree (and thus being free to vary) while not being part of other spanning trees (and thus being determined by the corresponding 'cycle' constraint). Our goal is to determine all encoding coefficients that are free to vary, while guaranteeing that encoding coefficients on the edges of the disconnected components have consistent values. In the conference version of this work [1] we proposed a greedy algorithm to find a spanning tree for each disconnected component that guarantees constancy among disconnected components. In this journal version we take a different approach.

We note that all disconnected components are isomorphic, that is, by an appropriate mapping of the labels in one component we get another component of the overall graph. For example, by replacing each 4 with a 5 in $\mathcal{G}^{(\{3,4\})}$ in Fig. 4a, we get $\mathcal{G}^{(\{3,5\})}$ in Fig. 4b. Based on this observation, we seek solutions for (27) in Theorem 2 where $\widetilde{\beta}^{(\mathcal{A})}_{\{k\}\cup\mathcal{T}}, \phi^{(\mathcal{A})}_{k,\mathcal{T}}, \mathsf{c}^{(\mathcal{A})}_{\mathcal{T}}$ do not depend on $\mathcal{A}$. This is possible if we can find an alternative formulation of the 'sign function' $\phi^{(\mathcal{A})}_{k,\mathcal{T}}$ in (27c) that does not depend on $\mathcal{A}$. In the case of our example, this amounts to show that we can change the sign of both $\alpha_{4,\{1\}}$ and $\alpha_{1,\{4\}}$ in $\mathcal{G}^{(\{4,5\})}$ in Fig. 4c and still have a scheme that guarantees we can successfully reconstruct $W_{\{4,5\}}$.

In the next subsection we show that indeed we can find an alternative formulation of the 'sign function' $\phi^{(\mathcal{A})}_{k,\mathcal{T}}$ in (27c) that does not depend on $\mathcal{A}$ and still guarantees successfully reconstruction of all non-sent multicast messages. This new 'sign function' is denoted by $\phi'_{k,\mathcal{T}}$ and is given by (30). With (30), for the case of our example, the overall graph is exactly as in Fig. 4 (where we had omitted the superscript $\mathcal{A}$) but with the sign of both $\alpha_{4,\{1\}}$ and $\alpha_{1,\{4\}}$ in $\mathcal{G}^{(\{4,5\})}$ in Fig. 4c flipped. This however is not a valid graph as the same vertex appears multiple components. What we do next is to 'merge' the different components into a single graph, which we refer to as the *universal graph* (as it captures all the constraints needed for successful reconstruction of all non-sent multicast messages). The universal graph is drawn as for the case case $\mathsf{K} - |\mathcal{L}| = t+1$ but by using the new 'sign function' in (30) to label the edges, i.e., in the universal graph an edge exists between $\widetilde{\beta}_{\{k\}\cup\mathcal{T}}$ and $\mathsf{c}_{\mathcal{T}}$ and has label $(-1)^{\phi'_{k,\mathcal{T}}}\alpha_{k,\mathcal{T}}$. For the case of our example, 'merging' the three components in Fig. 4 results in the universal graph in Fig. 5a.

**Remark 4.** When $\mathsf{K} - |\mathcal{L}| \geq t+1$, the disconnect components of the overall graph generated by Theorem 2 are subgraphs of the universal graph generated by Proposition 2, except possibly for the signs of some of the edge labels. Fig. 5 shows the three subgraphs of the universal graph corresponding to the 3 multicast messages that must be locally reconstructed. □

**Remark 5.** When $\mathsf{K} - |\mathcal{L}| \geq t + 1$, the number of encoding coefficients in the universal graph that are free to vary (i.e., are on a spanning tress) is $\binom{\mathsf{K}}{t} + \binom{\mathsf{K}}{t+1} - 1$. This is because the number of vertexes in the universal graph is $\binom{\mathsf{K}}{t} + \binom{\mathsf{K}}{t+1}$. Thus, the number of edges in a spanning tree on the universal graph is one less than the number of edges [19]. □

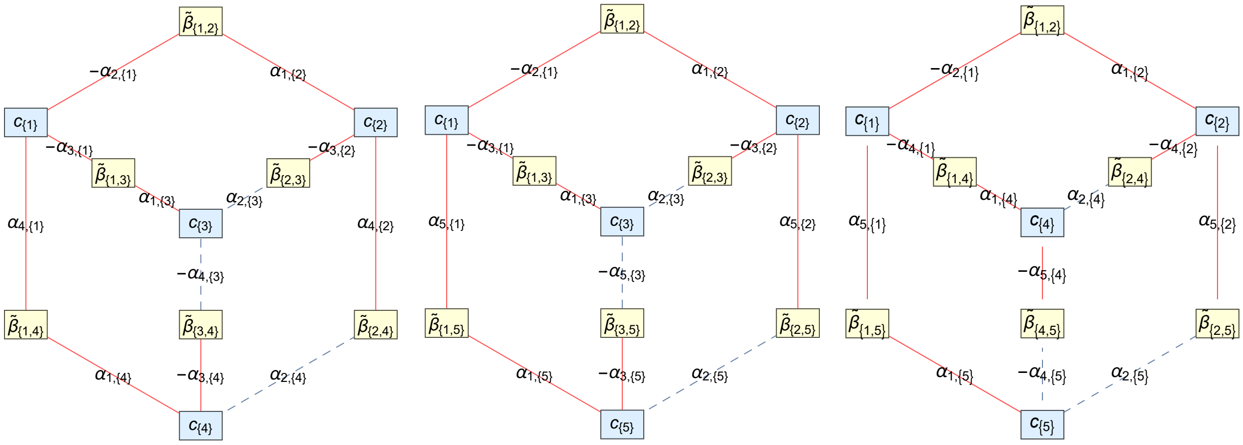
### *D. Proof of* (30) *– Proof of Proposition 2*

Graphs drawn as in Fig. 4 have cycles that represent constraints among encoding coefficients, and the shortest cycle has length 6. We refer to such a length-6 cycle as a *minimal cycle*, which has the following structure. Suppose $\mathcal{L}$ and $\mathcal{A}$ are defined as in the rest of the paper and are fixed. Assume $\mathcal{P} \in \Omega^{t-1}_{\mathcal{A}\cup\mathcal{L}}$ and $\{k_1, k_2, k_3\} \subseteq \mathcal{A} \cup \mathcal{L} \setminus \mathcal{P}$, where $k_1, k_2$ and $k_3$ are distinct. A minimal cycle involving $(\mathcal{P}, k_1, k_2, k_3)$ is shown in Fig. 6, where we used the shorthand notation

$$\rho_{k,\mathcal{T}} := (-1)^{\phi^{(\mathcal{A})}_{k,\mathcal{T}}} \cdot \alpha_{k,\mathcal{T}} \quad (60)$$

to label the edges. In Fig. 6, the red edges form a possible spanning tree and the left out edge (with label $\rho_{k_1,\{k_2\}\cup\mathcal{P}}$) has its value determined as

$$\rho_{k_1,\{k_2\}\cup\mathcal{P}} = \frac{\rho_{k_2,\{k_1\}\cup\mathcal{P}}\ \rho_{k_1,\{k_3\}\cup\mathcal{P}}\ \rho_{k_3,\{k_2\}\cup\mathcal{P}}}{\rho_{k_3,\{k_1\}\cup\mathcal{P}}\ \rho_{k_2,\{k_3\}\cup\mathcal{P}}}. \quad (61)$$



(a) Component $\mathcal{G}^{(\{3,4\})}$ for $W_{\{3,4\}}$ (b) Component $\mathcal{G}^{(\{3,5\})}$ for $W_{\{3,5\}}$ (c) Component $\mathcal{G}^{(\{4,5\})}$ for $W_{\{4,5\}}$

Fig. 4: The graph and possible spanning trees for the case $\mathsf{K}=5, r=2, t=1$. The convention is as in Fig. 3. For sake of legibility, we omitted the superscripts in the various sub-figures, which should be the index of the multicast message listed in the sub-caption.

In particular, starting from (61), in Appendix A we show that

$$\frac{(-1)^{\phi^{(\mathcal{A})}_{k_2,\{k_1\}\cup\mathcal{P}}}\ (-1)^{\phi^{(\mathcal{A})}_{k_1,\{k_3\}\cup\mathcal{P}}}\ (-1)^{\phi^{(\mathcal{A})}_{k_3,\{k_2\}\cup\mathcal{P}}}}{(-1)^{\phi^{(\mathcal{A})}_{k_3,\{k_1\}\cup\mathcal{P}}}\ (-1)^{\phi^{(\mathcal{A})}_{k_2,\{k_3\}\cup\mathcal{P}}}\ (-1)^{\phi^{(\mathcal{A})}_{k_1,\{k_2\}\cup\mathcal{P}}}} = -1, \quad (62)$$

which implies that

$$\frac{\alpha_{k_2,\{k_1\}\cup\mathcal{P}}\ \alpha_{k_1,\{k_3\}\cup\mathcal{P}}\ \alpha_{k_3,\{k_2\}\cup\mathcal{P}}}{\alpha_{k_3,\{k_1\}\cup\mathcal{P}}\ \alpha_{k_2,\{k_3\}\cup\mathcal{P}}\ \alpha_{k_1,\{k_2\}\cup\mathcal{P}}} = -1, \quad (63)$$

where (62) only includes the part "$(-1)^{\phi^{(\mathcal{A})}_{k,\mathcal{T}}}$" of the function $\rho_{k,\mathcal{T}}$ in (61), and (63) only the part "$\alpha_{k,\mathcal{T}}$." In Appendix A, we show that (62) holds by examining all the possible cases for $|\{k_1,k_2,k_3\}\cap\mathcal{L}|$.

It is easy to see that the new sign function $\phi'_{k,\mathcal{T}}$ in Proposition 2 also satisfies (62) and (61).

## VI. Extension to Private Function Retrieval (PFR)

So far we discussed the coded caching problem with scalar linear function retrieval and we showed the constraints a general YMA-type linear scheme must satisfy in order to guarantee successful decoding at all user. We next show how the derived framework applies to other problems as well.

The Private Information Retrieval problem (PIR) was first introduced by Chor *et al.* in [20]. It describes a scenario where a user aims to retrieve a single file from multiple non-colluding servers (all storing the same library of files) without reveling its desired file to any server. A trivial solution is to request all files from every server, but the network load would be extremely large. In [16], Sun and Jafar determined the capacity of PIR when the files are independent. Later, Sun and Jafar in [17] generalized the PIR setting to the case where there are linear dependency among the files, i.e., some files are linear combinations of the the remaining files. We refer to this setting as Private Function Retrieval (PFR). Surprisingly, [17] showed that the capacity of PFR is the same as that of PIR. The SLFR optimal solution under uncoded placement proposed by Wan *et. al.* in [18] (see Section II-C) was inspired by the PFR capacity achieving scheme in [17]: both schemes alternate the encoding coefficients between $-1$ and 1 in some controlled manner. In this section, we focus on the PFR scheme in [17], and show how our universal graph framework to determine a SLFR optimal linear scheme under uncoded placement extends to the PFR case.

### A. Problem Settings and Known Results

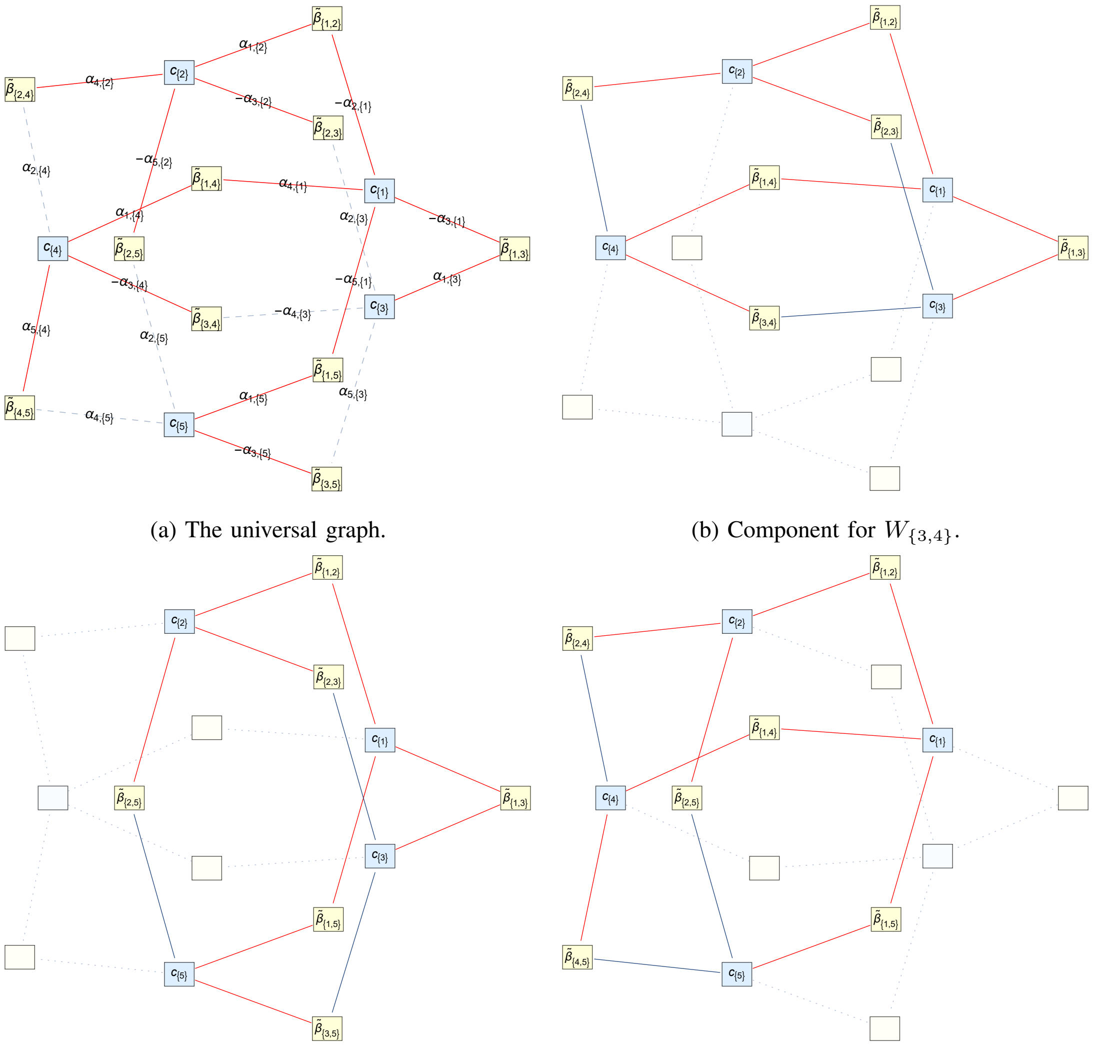
We start by restating the PFR problem formulation in [17] in the SLFR notation we have been using in this paper so far. In the following database is synonym of file. Suppose we have $\mathsf{K}$ databases in which $r$ of them are linearly independent (i.e., $\mathsf{K}\geq r$). Each independent database is comprised of $\mathsf{B}$ i.i.d. symbols uniformly distributed over a finite field $\mathbb{F}_\mathsf{q}$. Moreover, we have $\mathsf{S}$ non-colluding servers which store all databases. The user aims to retrieve the $\theta$-th database while preserving privacy toward any of the servers, i.e., by leaving all servers ignorant about the actual value $\theta$. The goal is to minimize the total number of symbols downloaded by user from servers.

In [17], Sun and Jafar purposed an achievable scheme where the user first downloads uncoded symbols per each database from each server. Then the user exploits the undesired downloaded symbols as side information to retrieve further desired symbols. The user repeats this procedure until the desired database is downloaded. In order to preserve privacy, the scheme must be symmetric across all serves and across all files. Each step of this scheme is actually equivalent to the delivery phase of an SLFR scheme; thus, the universal graph approach can be used to derive a general PFR linear scheme.

We are not going to describe the general optimal PFR scheme for all possible problem's parameters. Instead we give an example next to show how to use the universal graph approach in the PFR problem.

### B. Example A in [17]

Suppose we have $\mathsf{S}=2$ servers and $\mathsf{K}=4$ databases. The databases are denoted as $A,B,C,D$. We assume $A$



(a) The universal graph.

(b) Component for $W_{\{3,4\}}$.

(c) Component for $W_{\{3,5\}}$.

(d) Component for $W_{\{4,5\}}$.

Fig. 5: Example for $\mathsf{K} = 5, r = 2, t = 1$. Fig. 5a shows the universal graph and a spanning tree. Fig. 5b, Fig. 5c and Fig. 5d are the components of the universal graph showed in Fig. 4, and spanning trees obtained from the example spanning tree in Fig. 5a. For sake of legibility, all spanning trees are highlighted in red.

and $B$ are linearly independent, while $C$ and $D$ are linear combinations of $A$ and $B$. The $i$-th symbol of each database (i.e., $a_i, b_i, c_i, d_i$) can be expressed as

$$\begin{bmatrix} a_i \\ b_i \\ c_i \\ d_i \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & 0 \\ 0 & 1 \\ d_{3,1} & d_{3,2} \\ d_{4,1} & d_{4,2} \end{bmatrix}}_{\mathbb{D}} \begin{bmatrix} a_i \\ b_i \end{bmatrix}, \tag{64}$$

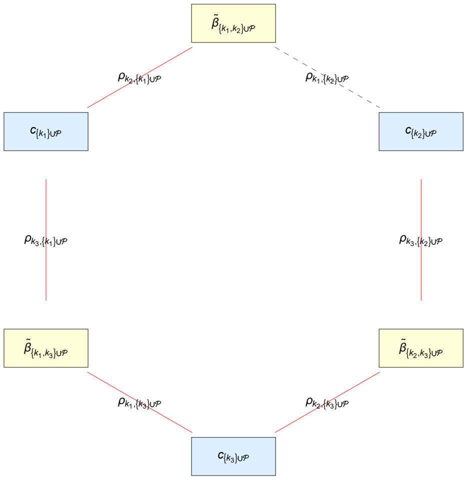
for some generator matrix $\mathbb{D}$. Here $r = \mathsf{rank}(\mathbb{D}) = 2$.

An optimal scheme is as follows [17]. When the user demands $A$, i.e., $\theta = 1$, the symbols dowloaded from the serves are

| $t$ | Server 1 | Server 2 |
|---|---|---|
| 0 | $a_1, b_1, c_1, d_1$ | $a_2, b_2, c_2, d_2$ |
| 1 | $a_3 - b_2$<br>$a_4 - c_2$<br>$a_5 - d_2$<br>$b_4 - c_3$<br>$b_5 - d_3$<br>$c_5 - d_4$ | $a_6 - b_1$<br>$a_7 - c_1$<br>$a_8 - d_1$<br>$b_7 - c_6$<br>$b_8 - d_6$<br>$c_8 - d_7$ |
| 2 | $a_9 - b_7 + c_6$<br>$a_{10} - b_8 + d_6$<br>$a_{11} - c_8 + d_7$<br>$b_{11} - c_{10} + d_9$ | $a_{12} - b_4 + c_3$<br>$a_{13} - b_5 + d_3$<br>$a_{14} - c_5 + d_4$<br>$b_{14} - c_{13} + d_{12}$ |
| 3 | $a_{15} - b_{14} + c_{13} - d_{12}$ | $a_{16} - b_{11} + c_{10} - d_9$ |

(65)

where we highlight the in cyan the 'redundant' messages , that is, the user can locally construct those rather than retrieving them from the servers. For example, the pair $(c_1, d_1)$ is a function of the pair $(a_1, b_1)$ through the linear transformation in (64).



$\tilde{\beta}_{\{k_1,k_2\}\cup\mathcal{P}}$
$\rho_{k_2,\{k_1\}\cup\mathcal{P}}$
$\rho_{k_1,\{k_2\}\cup\mathcal{P}}$
$c_{\{k_1\}\cup\mathcal{P}}$
$c_{\{k_2\}\cup\mathcal{P}}$
$\rho_{k_3,\{k_1\}\cup\mathcal{P}}$
$\rho_{k_3,\{k_2\}\cup\mathcal{P}}$
$\tilde{\beta}_{\{k_1,k_3\}\cup\mathcal{P}}$
$\tilde{\beta}_{\{k_2,k_3\}\cup\mathcal{P}}$
$\rho_{k_1,\{k_3\}\cup\mathcal{P}}$
$\rho_{k_2,\{k_3\}\cup\mathcal{P}}$
$c_{\{k_3\}\cup\mathcal{P}}$

Fig. 6: A minimal cycle. Fixed $\mathcal{L}$ and $\mathcal{A}$, assume $\mathcal{P} \in \Omega^{t-1}_{\mathcal{A}\cup\mathcal{L}}$ and $\{k_1, k_2, k_3\} \subseteq \mathcal{A} \cup \mathcal{L} \setminus \mathcal{P}$. $\rho_{k,\mathcal{T}} := (-1)^{\phi^{(\mathcal{A})}_{k,\mathcal{T}}} \alpha_{k,\mathcal{T}}$. For the sake of legibility, we omitted all superscript $\mathcal{A}$ in all vertexes.

We see that with $\mathsf{S} = 2$ servers, the messages in each block $t$ in (65) sent by the same server are equivalent to the multicast messages generated as in the $(\mathsf{K}, r, \mathsf{q})$ SLFR problem with memory size $t$ and with demand matrix $\mathbb{D}$ in (64). For example, one can see that

- the messages corresponding to the 'row' $t = 1$ in (65) sent by Server 1 are the same as the multicast messages of an SLFR problem with 4 users, 2 leaders, cache contents

$$Z_1 = \{a_2, b_2, c_2, d_2\}, \tag{66a}$$
$$Z_2 = \{a_3, b_3, c_3, d_3\}, \tag{66b}$$
$$Z_3 = \{a_4, b_4, c_4, d_4\}, \tag{66c}$$
$$Z_4 = \{a_5, b_5, c_5, d_5\}, \tag{66d}$$

  and with demand matrix $\mathbb{D}$ in (64). The correspondence between PFR and SLFR is evident if we map the symbol indexes in (66) to the subsets in $\Omega^1_{[4]}$ according to

$$(2, 3, 4, 5) \to (\{1\}, \{2\}, \{3\}, \{4\}). \tag{67}$$

  For example, "$2 \to \{1\}$" means that symbol $a_2$ is cached by SLFR user 1, in other words, $a_2$ in PFR would be indicated as $a_{\{1\}}$ in SLFR. Each SLFR user demands symbols cached by other SLFR users; in this sense, the SLFR scheme generates multicast messages exactly same as the PFR.
- the messages in the 'row' $t = 2$ in (65) sent by Server 2 are the same as the multicast messages of an SLFR problem with 4 users, 2 leaders, cache contents

$$Z_1 = \bigcup_{i\in\mathcal{I}_1} \{a_i, b_i, c_i, d_i\}, \text{ with } \mathcal{I}_1 = \{3, 4, 5\}, \tag{68a}$$
$$Z_2 = \bigcup_{i\in\mathcal{I}_2} \{a_i, b_i, c_i, d_i\}, \text{ with } \mathcal{I}_2 = \{3, 12, 13\}, \tag{68b}$$
$$Z_3 = \bigcup_{i\in\mathcal{I}_3} \{a_i, b_i, c_i, d_i\}, \text{ with } \mathcal{I}_3 = \{4, 12, 14\}, \tag{68c}$$
$$Z_4 = \bigcup_{i\in\mathcal{I}_4} \{a_i, b_i, c_i, d_i\}, \text{ with } \mathcal{I}_4 = \{5, 13, 14\}, \tag{68d}$$

  and with demand matrix $\mathbb{D}$ in (64). The correspondence between PFR and SLFR is again evident if we map the symbol indexes in (68) to the subsets $\Omega^2_{[4]}$ as follows

$$\begin{aligned}&(3, 4, 5, 12, 13, 14)\\ &\to \big(\{1, 2\}, \{1, 3\}, \{1, 4\}, \{2, 3\}, \{2, 4\}, \{3, 4\}\big).\end{aligned} \tag{69}$$

  For example, "$3 \to \{1, 2\}$" means that symbol $a_3$ is cached by SLFR users 1 and 2, in other words, $a_3$ in PFR would be indicated as $a_{\{1,2\}}$ in SLFR. In this sense, it is straightforward that the PFR messages are equivalent to SLFR multicast messages.

### C. Constraints among Coefficients of Symbols

In the PFR scheme in [17], the encoding coefficients alternate between $+1$ and $-1$. With the help of universal graph, it is possible to select other elements in $\mathbb{F}_\mathsf{q}$ as encoding coefficients and capture the constraints among them. The PFR user exploits previous messages which do not contain desired symbols as side information for future messages sent by another server. We explained above how those PFR messages are equivalent to SLFR multicast messages.

We denote the encoding coefficient with the SLFR-like notation $\{\alpha_{i,\mathcal{W}} \in \mathbb{F}_\mathsf{q} \setminus \{0\}, i \in [\mathsf{K}], \mathcal{W} \subseteq [\mathsf{K}]\}$ by following the maps in (67) and in (69). The most general version of the scheme in (65) when $\theta = 1$ is requested is as shows in (70) at the top of the next page. In (70), we highlighted in magenta the symbols treated as side information for further messages, and in blue the symbols that are side information in the current block. For example, a magenta symbol from Server 2 in 'row' $t = 1$ is a blue symbol for Server 1 in 'row' $t = 2$, possibly multiplied by an encoding coefficient. Encoding coefficients with modulo different from one will not affect privacy.

In (70), we can build a universal graph for the messages sent by the same server in each 'row' $t \in [\mathsf{K} - r - 1]$, i.e., when redundant multicast messages exist in the equivalent SLFR problem. However, there are some additional equality constraints in PFR among the encoding coefficients not present in SLFR, that is, the messages highlighted in blue in a 'row' should be equal to the messages highlighted in magenta in previous 'row'. For example, we should have constraints such as $\alpha_{2,\{3\}} = \alpha_{2,\{1,3\}}$, $\alpha_{3,\{2\}} = \alpha_{3,\{1,2\}}$, etc. A solution to this new issue is as follows: the encoding coefficients that are not-free in the universal graph for the messages sent by the same server in 'row' $t$ will be inherited by the universal graph in 'row' $t + 1$ for the other server. Therefore, one can derive the conditions a general linear scheme for PFR must satisfy by solving a series of spanning tree problems for the various universal graphs in the corresponding SLFR problem, one such problem for each 'row' at each server.

## VII. Conclusion

In this paper, we investigated the constraints that a linear scheme for cache-aided scalar linear function retrieval must satisfy in order to be feasible. We showed that the constraints

| $t$ | Server 1 | Server 2 |
|---|---|---|
| 0 | | $a_2, b_2, c_2, d_2$ |
| 1 | $\alpha_{1,\{2\}}a_3 + \alpha_{2,\{1\}}b_2$<br>$\alpha_{1,\{3\}}a_4 + \alpha_{3,\{1\}}c_2$<br>$\alpha_{1,\{4\}}a_5 + \alpha_{4,\{1\}}d_2$<br>$\alpha_{2,\{3\}}b_4 + \alpha_{3,\{2\}}c_3$<br>$\alpha_{2,\{4\}}b_5 + \alpha_{4,\{2\}}d_3$<br>$\alpha_{3,\{4\}}c_5 + \alpha_{4,\{3\}}d_4$ | |
| 2 | | $\alpha_{1,\{2,3\}}a_{12} + \alpha_{2,\{1,3\}}b_4 + \alpha_{3,\{1,2\}}c_3$<br>$\alpha_{1,\{2,4\}}a_{13} + \alpha_{2,\{1,4\}}b_5 + \alpha_{4,\{1,2\}}d_3$<br>$\alpha_{1,\{3,4\}}a_{14} + \alpha_{3,\{1,4\}}c_5 + \alpha_{4,\{1,3\}}d_4$<br>$\alpha_{2,\{3,4\}}b_{14} + \alpha_{3,\{2,4\}}c_{13} + \alpha_{4,\{2,3\}}d_{12}$ |
| 3 | $\alpha_{1,\{2,3,4\}}a_{15} + \alpha_{2,\{1,3,4\}}b_{14}+$<br>$+\alpha_{3,\{1,2,4\}}c_{13} + \alpha_{4,\{1,2,3\}}d_{12}$ | |

(70)

among the parameters of a feasible linear scheme are captured by the cycles of the universal graph. Equivalently, we showed that a spanning tree for the universal graph identifies all the parameters of the feasible linear scheme that are free to vary. The structure of our general scheme sheds light into a scheme that had been previously proposed in the literature and naturally extends to problems such as private function computation/retrieval.

## Appendix A
## Proof of (62) and (63)

The condition in (62) is equivalent to

$$\begin{aligned}&\phi^{(\mathcal{A})}_{k_2,\{k_1\}\cup\mathcal{P}} + \phi^{(\mathcal{A})}_{k_1,\{k_3\}\cup\mathcal{P}} + \phi^{(\mathcal{A})}_{k_3,\{k_2\}\cup\mathcal{P}} - \phi^{(\mathcal{A})}_{k_3,\{k_1\}\cup\mathcal{P}}\\ &\quad - \phi^{(\mathcal{A})}_{k_2,\{k_3\}\cup\mathcal{P}} - \phi^{(\mathcal{A})}_{k_1,\{k_2\}\cup\mathcal{P}} = \textit{(An odd number)},\end{aligned} \tag{71}$$

which is equivalent to

$$\begin{aligned}&\phi^{(\mathcal{A})}_{k_2,\{k_1\}\cup\mathcal{P}} \pm \phi^{(\mathcal{A})}_{k_1,\{k_3\}\cup\mathcal{P}} \pm \phi^{(\mathcal{A})}_{k_3,\{k_2\}\cup\mathcal{P}} \pm \phi^{(\mathcal{A})}_{k_3,\{k_1\}\cup\mathcal{P}}\\ &\quad \pm \phi^{(\mathcal{A})}_{k_2,\{k_3\}\cup\mathcal{P}} \pm \phi^{(\mathcal{A})}_{k_1,\{k_2\}\cup\mathcal{P}} = \textit{(An odd number)},\end{aligned} \tag{72}$$

since neither plus nor minus affects parity. We show here that (72) is true. We separate the proof into four cases, i.e., $|\{k_1, k_2, k_3\} \cap \mathcal{L}| \in \{0, 1, 2, 3\}$. Assume $k_1 < k_2 < k_3$. By definition of the function $\phi^{(\mathcal{A})}_{k,\mathcal{T}}$ we have

1) $k_1, k_2, k_3$ are non-leader users:

$$\begin{aligned}&\phi^{(\mathcal{A})}_{k_1,\{k_3\}\cup\mathcal{P}} + \phi^{(\mathcal{A})}_{k_1,\{k_2\}\cup\mathcal{P}}\\ &= \mathsf{Ind}_{\mathcal{A}\setminus(\{k_3\}\cup\mathcal{P}),k_1} + \mathsf{Ind}_{\mathcal{A}\setminus(\{k_2\}\cup\mathcal{P}),k_1}\\ &= \mathsf{Ind}_{\mathcal{A}\setminus\mathcal{P},k_1} + \mathsf{Ind}_{\mathcal{A}\setminus\mathcal{P},k_1} = \textit{(even)},\end{aligned} \tag{73a}$$

$$\begin{aligned}&\phi^{(\mathcal{A})}_{k_2,\{k_1\}\cup\mathcal{P}} + \phi^{(\mathcal{A})}_{k_2,\{k_3\}\cup\mathcal{P}}\\ &= \mathsf{Ind}_{\mathcal{A}\setminus(\{k_1\}\cup\mathcal{P}),k_2} + \mathsf{Ind}_{\mathcal{A}\setminus(\{k_3\}\cup\mathcal{P}),k_2}\\ &= (\mathsf{Ind}_{\mathcal{A}\setminus\mathcal{P},k_2} - 1) + \mathsf{Ind}_{\mathcal{A}\setminus\mathcal{P},k_2} = \textit{(odd)},\end{aligned} \tag{73b}$$

$$\begin{aligned}&\phi^{(\mathcal{A})}_{k_3,\{k_2\}\cup\mathcal{P}} + \phi^{(\mathcal{A})}_{k_3,\{k_1\}\cup\mathcal{P}}\\ &= \mathsf{Ind}_{\mathcal{A}\setminus(\{k_2\}\cup\mathcal{P}),k_3} + \mathsf{Ind}_{\mathcal{A}\setminus(\{k_1\}\cup\mathcal{P}),k_3}\\ &= (\mathsf{Ind}_{\mathcal{A}\setminus\mathcal{P},k_3} - 1) + (\mathsf{Ind}_{\mathcal{A}\setminus\mathcal{P},k_3} - 1)\\ &= \textit{(even)}.\end{aligned} \tag{73c}$$

Therefore, the sum of the above three terms is an odd number.

2) $k_1$ is a leader:

$$\begin{aligned}&\phi^{(\mathcal{A})}_{k_1,\{k_3\}\cup\mathcal{P}} + \phi^{(\mathcal{A})}_{k_1,\{k_2\}\cup\mathcal{P}}\\ &= (1 + \mathsf{Ind}_{(\{k_1,k_3\}\cup\mathcal{P})\setminus\mathcal{A},k_1}) + (1 + \mathsf{Ind}_{(\{k_1,k_2\}\cup\mathcal{P})\setminus\mathcal{A},k_1})\\ &= (1 + \mathsf{Ind}_{(\{k_1\}\cup\mathcal{P})\setminus\mathcal{A},k_1}) + (1 + \mathsf{Ind}_{(\{k_1\}\cup\mathcal{P})\setminus\mathcal{A},k_1})\\ &= \textit{(even)},\end{aligned} \tag{74a}$$

$$\begin{aligned}&\phi^{(\mathcal{A})}_{k_2,\{k_1\}\cup\mathcal{P}} + \phi^{(\mathcal{A})}_{k_2,\{k_3\}\cup\mathcal{P}}\\ &= \mathsf{Ind}_{\mathcal{A}\setminus(\{k_1\}\cup\mathcal{P}),k_2} + \mathsf{Ind}_{\mathcal{A}\setminus(\{k_3\}\cup\mathcal{P}),k_2}\\ &= \mathsf{Ind}_{\mathcal{A}\setminus\mathcal{P},k_2} + \mathsf{Ind}_{\mathcal{A}\setminus\mathcal{P},k_2} = \textit{(even)},\end{aligned} \tag{74b}$$

$$\begin{aligned}&\phi^{(\mathcal{A})}_{k_3,\{k_2\}\cup\mathcal{P}} + \phi^{(\mathcal{A})}_{k_3,\{k_1\}\cup\mathcal{P}}\\ &= \mathsf{Ind}_{\mathcal{A}\setminus(\{k_2\}\cup\mathcal{P}),k_3} + \mathsf{Ind}_{\mathcal{A}\setminus(\{k_1\}\cup\mathcal{P}),k_3}\\ &= (\mathsf{Ind}_{\mathcal{A}\setminus\mathcal{P},k_3} - 1) + \mathsf{Ind}_{\mathcal{A}\setminus\mathcal{P},k_3} = \textit{(odd)}.\end{aligned} \tag{74c}$$

Therefore, the sum of the above three terms is an odd number.

3) $k_1, k_2$ are leader users:

$$\begin{aligned}&\phi^{(\mathcal{A})}_{k_1,\{k_3\}\cup\mathcal{P}} + \phi^{(\mathcal{A})}_{k_1,\{k_2\}\cup\mathcal{P}}\\ &= (1 + \mathsf{Ind}_{(\{k_1,k_3\}\cup\mathcal{P})\setminus\mathcal{A},k_1}) + (1 + \mathsf{Ind}_{(\{k_1,k_2\}\cup\mathcal{P})\setminus\mathcal{A},k_1})\\ &= (1 + \mathsf{Ind}_{(\{k_1\}\cup\mathcal{P})\setminus\mathcal{A},k_1}) + (1 + \mathsf{Ind}_{(\{k_1\}\cup\mathcal{P})\setminus\mathcal{A},k_1})\\ &= \textit{(even)},\end{aligned} \tag{75a}$$

$$\begin{aligned}&\phi^{(\mathcal{A})}_{k_2,\{k_1\}\cup\mathcal{P}} + \phi^{(\mathcal{A})}_{k_2,\{k_3\}\cup\mathcal{P}}\\ &= (1 + \mathsf{Ind}_{(\{k_1,k_2\}\cup\mathcal{P})\setminus\mathcal{A},k_2}) + (1 + \mathsf{Ind}_{(\{k_2,k_3\}\cup\mathcal{P})\setminus\mathcal{A},k_2})\\ &= (2 + \mathsf{Ind}_{(\{k_2\}\cup\mathcal{P})\setminus\mathcal{A},k_2}) + (1 + \mathsf{Ind}_{(\{k_2\}\cup\mathcal{P})\setminus\mathcal{A},k_2})\\ &= \textit{(odd)},\end{aligned} \tag{75b}$$

$$\begin{aligned}&\phi^{(\mathcal{A})}_{k_3,\{k_2\}\cup\mathcal{P}} + \phi^{(\mathcal{A})}_{k_3,\{k_1\}\cup\mathcal{P}}\\ &= \mathsf{Ind}_{\mathcal{A}\setminus(\{k_2\}\cup\mathcal{P}),k_3} + \mathsf{Ind}_{\mathcal{A}\setminus(\{k_1\}\cup\mathcal{P}),k_3}\\ &= \mathsf{Ind}_{\mathcal{A}\setminus\mathcal{P},k_3} + \mathsf{Ind}_{\mathcal{A}\setminus\mathcal{P},k_3} = \textit{(even)}.\end{aligned} \tag{75c}$$

Therefore, the sum of the above three terms is an odd number.

4) $k_1, k_2, k_3$ are leader users:

$$\begin{aligned}&\phi^{(\mathcal{A})}_{k_1,\{k_3\}\cup\mathcal{P}} + \phi^{(\mathcal{A})}_{k_1,\{k_2\}\cup\mathcal{P}}\\ &= (1 + \mathsf{Ind}_{(\{k_1,k_3\}\cup\mathcal{P})\setminus\mathcal{A},k_1}) + (1 + \mathsf{Ind}_{(\{k_1,k_2\}\cup\mathcal{P})\setminus\mathcal{A},k_1})\\ &= (1 + \mathsf{Ind}_{(\{k_1\}\cup\mathcal{P})\setminus\mathcal{A},k_1}) + (1 + \mathsf{Ind}_{(\{k_1\}\cup\mathcal{P})\setminus\mathcal{A},k_1})\\ &= \textit{(even)},\end{aligned} \tag{76a}$$

$$\phi^{(\mathcal{A})}_{k_2,\{k_1\}\cup\mathcal{P}} + \phi^{(\mathcal{A})}_{k_2,\{k_3\}\cup\mathcal{P}}$$

$$
\begin{aligned}
&= (1 + \mathsf{Ind}_{(\{k_1,k_2\}\cup\mathcal{P})\setminus\mathcal{A},k_2}) + (1 + \mathsf{Ind}_{(\{k_2,k_3\}\cup\mathcal{P})\setminus\mathcal{A},k_2}) \\
&= (2 + \mathsf{Ind}_{(\{k_2\}\cup\mathcal{P})\setminus\mathcal{A},k_2}) + (1 + \mathsf{Ind}_{(\{k_2\}\cup\mathcal{P})\setminus\mathcal{A},k_2}) \\
&= \textit{(odd)}, \qquad (76b) \\
&\phi^{(\mathcal{A})}_{k_3,\{k_2\}\cup\mathcal{P}} + \phi^{(\mathcal{A})}_{k_3,\{k_1\}\cup\mathcal{P}} \\
&= (1 + \mathsf{Ind}_{(\{k_2,k_3\}\cup\mathcal{P})\setminus\mathcal{A},k_3}) + (1 + \mathsf{Ind}_{(\{k_1,k_3\}\cup\mathcal{P})\setminus\mathcal{A},k_3}) \\
&= (2 + \mathsf{Ind}_{(\{k_3\}\cup\mathcal{P})\setminus\mathcal{A},k_3}) + (2 + \mathsf{Ind}_{(\{k_3\}\cup\mathcal{P})\setminus\mathcal{A},k_3}) \\
&= \textit{(even)}. \qquad (76c)
\end{aligned}
$$

Therefore, the sum of the above three terms is an odd number.

This concludes the proof that the condition in (62) is true. With (62) into (61), we have that the condition in (63) is also true.

## References

[1] Y. Ma and D. Tuninetti, "A general coded caching scheme for scalar linear function retrieval," in *2021 IEEE International Symposium on Information Theory (ISIT)*, pp. 2816–2821, IEEE, 2021.

[2] M. A. Maddah-Ali and U. Niesen, "Fundamental limits of caching," *IEEE Transactions on Information Theory*, vol. 60, no. 5, pp. 2856–2867, 2014.

[3] Q. Yu, M. A. Maddah-Ali, and A. S. Avestimehr, "The exact rate-memory tradeoff for caching with uncoded prefetching," *IEEE Transactions on Information Theory*, vol. 64, no. 2, pp. 1281–1296, 2017.

[4] K. Wan, D. Tuninetti, and P. Piantanida, "An index coding approach to caching with uncoded cache placement," *IEEE Transactions on Information Theory*, vol. 66, no. 3, pp. 1318–1332, 2020.

[5] A. M. Ibrahim, A. A. Zewail, and A. Yener, "Benefits of edge caching with coded placement for asymmetric networks and shared caches," *IEEE Journal on Selected Areas in Information Theory*, vol. 2, no. 4, pp. 1240–1252, 2021.

[6] Q. Yu, M. A. Maddah-Ali, and A. S. Avestimehr, "Characterizing the rate-memory tradeoff in cache networks within a factor of 2," *IEEE Transactions on Information Theory*, vol. 65, no. 1, pp. 647–663, 2018.

[7] A. Sengupta, R. Tandon, and T. C. Clancy, "Fundamental limits of caching with secure delivery," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 2, pp. 355–370, 2014.

[8] K. Wan and G. Caire, "On coded caching with private demands," *IEEE Transactions on Information Theory*, vol. 67, no. 1, pp. 358–372, 2020.

[9] M. Cheng, Y. Li, X. Zhong, and R. Wei, "Improved constructions of coded caching schemes for combination networks," *IEEE Transactions on Communications*, vol. 68, no. 10, pp. 5965–5975, 2020.

[10] K. Wan, D. Tuninetti, M. Ji, and P. Piantanida, "Combination networks with end-user-caches: Novel achievable and converse bounds under uncoded cache placement," *arXiv preprint arXiv:1701.06884*, 2017.

[11] M. Ji, G. Caire, and A. F. Molisch, "Fundamental limits of caching in wireless d2d networks," *IEEE Transactions on Information Theory*, vol. 62, no. 2, pp. 849–869, 2015.

[12] S. Li, M. A. Maddah-Ali, Q. Yu, and A. S. Avestimehr, "A fundamental tradeoff between computation and communication in distributed computing," *IEEE Transactions on Information Theory*, vol. 64, no. 1, pp. 109–128, 2017.

[13] M. A. Attia and R. Tandon, "Near optimal coded data shuffling for distributed learning," *IEEE Transactions on Information Theory*, vol. 65, no. 11, pp. 7325–7349, 2019.

[14] A. Elmahdy and S. Mohajer, "On the fundamental limits of coded data shuffling for distributed machine learning," *IEEE Transactions on Information Theory*, vol. 66, no. 5, pp. 3098–3131, 2020.

[15] K. Wan, D. Tuninetti, M. Ji, G. Caire, and P. Piantanida, "Fundamental limits of decentralized data shuffling," *IEEE Transactions on Information Theory*, vol. 66, no. 6, pp. 3616–3637, 2020.

[16] H. Sun and S. A. Jafar, "The capacity of private information retrieval," *IEEE Transactions on Information Theory*, vol. 63, no. 7, pp. 4075–4088, 2017.

[17] H. Sun and S. A. Jafar, "The capacity of private computation," *IEEE Transactions on Information Theory*, vol. 65, no. 6, pp. 3880–3897, 2018.

[18] K. Wan, H. Sun, M. Ji, D. Tuninetti, and G. Caire, "On the optimal load-memory tradeoff of cache-aided scalar linear function retrieval," *IEEE Transactions on Information Theory*, vol. 67, no. 6, pp. 4001–4018, 2021.

[19] J. Kleinberg and E. Tardos, *Algorithm design*. Pearson Education, 2006.

[20] B. Chor, O. Goldreich, E. Kushilevitz, and M. Sudan, "Private information retrieval," in *Proceedings of IEEE 36th Annual Foundations of Computer Science*, pp. 41–50, IEEE, 1995.

**Yinbin Ma** (Student Member, IEEE) is a Ph.D. candidate in Electrical and Computer Engineering (ECE) at University of Illinois Chicago (UIC). He received a Master's degree of Science in ECE from UIC in 2020, and a Bachelor's degree of Science in Computer Science from Xidian University in 2019. His research interests include the distributed information management and information theory.

**Daniela Tuninetti** (Fellow, IEEE) received the Ph.D. degree in Electrical Engineering from ENST/Télécom ParisTech, Paris, France, in 2002, with work done at the Eurecom Institute, Sophia Antipolis, France. She is currently a Professor and Department Head of Electrical and Computer Engineering (ECE) at University of Illinois Chicago (UIC), where she joined in 2005. She was a Post-Doctoral Research Associate with the School of Communication and Computer Science, Swiss Federal Institute of Technology in Lausanne (EPFL), Lausanne, Switzerland, from 2002 to 2004. Her research interests include the ultimate performance limits of wireless interference networks (with special emphasis on cognition and user cooperation), coexistence between radar and communication systems, multi-relay networks, content-type coding, cache-aided systems, and distributed private coded computing. She was a recipient of the Best Paper Award at the European Wireless Conference in 2002, the NSF CAREER Award in 2007, and named as the University of Illinois Scholar in 2015. She was the Editor-in-Chief of the IEEE Information Theory Society Newsletter from 2006 to 2008, an Editor of IEEE COMMUNICATIONS LETTERS from 2006 to 2009, IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS from 2011 to 2014, and IEEE TRANSACTIONS ON INFORMATION THEORY from 2014 to 2017. She is an Editor of IEEE TRANSACTIONS ON COMMUNICATIONS. She is currently a Distinguished Lecturer and an elected member of the Board of Governors of the IEEE Information Theory Society.