On the Fundamental Limits of Coded Caching With Correlated Files of Combinatorial Overlaps

Kai Wan, *Member, IEEE*, Daniela Tuninetti, *Fellow, IEEE*, Mingyue Ji, *Member, IEEE*, and Giuseppe Caire, *Fellow, IEEE*

***Abstract*— This paper studies the fundamental limits of the shared-link coded caching problem with correlated files, where a server with a library of N files communicates with K users who can locally cache M files. Given an integer $\mathsf{r} \in [\mathsf{N}]$, correlation is modelled as follows: each r-subset of files contains a unique common block. The tradeoff between the cache size and the average transmitted load over the uniform demand distribution is studied. First, a converse bound under the constraint of uncoded cache placement (i.e., each user directly stores a subset of the library bits) is derived. Then, a caching scheme for the case where every user demands a distinct file (possible for $\mathsf{N} \geq \mathsf{K}$) is shown to be optimal under the constraint of uncoded cache placement. This caching scheme is further proved to be decodable and optimal under the constraint of uncoded cache placement when (i) $\mathsf{KrM} \leq 2\mathsf{N}$ or $\mathsf{KrM} \geq (\mathsf{K}-1)\mathsf{N}$ or $\mathsf{r} \in \{1, 2, \mathsf{N}-1, \mathsf{N}\}$, and (ii) when the number of distinct demanded files is no larger than four. Finally, a new delivery scheme based on interference alignment which jointly serves the users' demands is shown to be order optimal to within a factor of 2 under the constraint of uncoded cache placement. As an extension, the above exact and order optimal results can be extended to the worst-case load. As by-products, an extension of the proposed scheme for $\mathsf{M} = \mathsf{N}/\mathsf{K}$ is shown to reduce the load of state-of-the-art schemes for the coded caching problem where the users can request multiple files; the proposed scheme for distinct demands can be extended to the coded distributed computing problem with a central server, which achieves the optimal transmission load over the binary field.**

***Index Terms*— Coded caching, correlated files, interference alignment.**

Manuscript received 12 August 2022; revised 29 January 2023; accepted 15 June 2023. Date of publication 30 June 2023; date of current version 15 September 2023. The work of Kai Wan was supported in part by the National Natural Science Foundation of China under Grant NSFC-12141107. The work of Daniela Tuninetti was supported in part by NSF under Award 1910309. The work of Mingyue Ji was supported in part by NSF under Award 1817154 and Award 1824558 and in part by NSF CAREER under Grant 2145835. The work of Giuseppe Caire was supported in part by the European Research Council under the ERC Advanced Grant through CARENET under Grant 789190. An earlier version of this paper was presented in part at the 2019 IEEE International Symposium on Information Theory [DOI: 10.1109/ISIT.2019.8849314]. *(Corresponding author: Kai Wan.)*

Kai Wan was with the Electrical Engineering and Computer Science Department, Technische Universität Berlin, 10587 Berlin, Germany. He is now with the School of Electronic Information and Communications, Huazhong University of Science and Technology, Wuhan 430074, China (e-mail: kai_wan@hust.edu.cn).

Daniela Tuninetti is with the Electrical and Computer Engineering Department, University of Illinois Chicago, Chicago, IL 60607 USA (e-mail: danielat@uic.edu).

Mingyue Ji is with the Electrical and Computer Engineering Department, The University of Utah, Salt Lake City, UT 84112 USA (e-mail: mingyue.ji@utah.edu).

Giuseppe Caire is with the Electrical Engineering and Computer Science Department, Technische Universität Berlin, 10587 Berlin, Germany (e-mail: caire@tu-berlin.de).

Communicated by G. Ge, Associate Editor for Coding and Decoding.

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TIT.2023.3291216.

Digital Object Identifier 10.1109/TIT.2023.3291216

I. Introduction

CACHE is a network component that leverages the device memory to store data so that future requests for that data can be served faster. Two phases are included in a caching system: i) the cache placement phase: content is pushed into each cache without knowledge of future demands; ii) the delivery phase: after each user has made its request and according to the cache contents, the server transmits coded packets in order to satisfy the users' demands. As in the classical setting in [2], we consider that the placement is performed offline, and the goal is to minimize the number of transmitted bits (load) from the server to the users during the delivery phase.

Information theoretic coded caching was originally proposed by Maddah-Ali and Niesen (MAN) in [2] for a shared-link caching system containing a server with a library of N equal-length files, connected to K users through a noiseless shared link. Each user can store M files in its local cache without knowledge of later demands. In the delivery phase, each user demands one file. The MAN scheme uses a combinatorial design in the cache placement phase such that, during delivery, multicast messages simultaneously satisfy the demands of different users. Under the constraint of uncoded cache placement (i.e., each user directly caches a subset of the library bits) and for worst-case load, the MAN scheme was proved to be optimal when $\mathsf{N} \geq \mathsf{K}$ [3], [4]. On the observation that some MAN linear combinations are redundant if there exist files demanded by several users, the authors in [4] improved the MAN delivery scheme and characterized the optimal worst-case load (and also the average load over the uniform demand distribution) under the constraint of uncoded cache placement for any K, N. The same authors proved in [5] that the multiplicative gap between the optimal caching scheme with uncoded cache placement and any caching scheme with coded cache placement is at most 2.

Coded caching strategies have been applied to several different models, such as decentralized systems [6], device-to-device (D2D) systems [7], topological networks [8], [9], [10], and various types of demands, such as linear functions [11], [12], matrix multiplication [13], secure demands [14], private

0018-9448 © 2023 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See https://www.ieee.org/publications/rights/index.html for more information.

Authorized licensed use limited to: University of Illinois at Chicago Library. Downloaded on November 08,2023 at 23:24:46 UTC from IEEE Xplore. Restrictions apply.

demands [15], [16]. The above works assume that the file in the library are mutually independent; i.e., they are realizations of statistically independent random variables. However, in practice there may be some correlations among different files (e.g., videos, image streams, etc.). Coded caching with correlated files was originally considered in [17], where correlation-aware coded caching schemes were proposed. In this work, we consider a coded caching problem with *correlated files*, where different files have common parts (i.e., overlaps).[1] In the following, we will review the literature of coded caching with correlated files, and then introduce our main contributions in this paper.

A. Related Previous Works

1) Coded Caching With Correlated Files: In [17], the files are divided into two sets, referred to as I-files and P-files, where the I-files are composed of some entire files and P-files are composed of inter-compressed files with respect to their corresponding I-files. By treating the delivery phase as an index coding problem with multiple requests, the authors in [17] proposed a delivery scheme based on graph coloring. In [19], after the Gray-Wyner source coding, the authors modelled correlation as each subset of files has an exclusively common part, which is then treated as an independent block/file in a coded multicast problem. Caching schemes for two-file K-user system (proved to be optimal) and three-file two-user system (optimal for the large cache case) were proposed. In [21], the caching problem with correlated files of combinatorial overlaps, where the length of the common part among each $\ell \in \{1, \ldots, \mathsf{N}\}$ files (referred to as a 'ℓ-block') is the same, was considered; each file contains $\binom{\mathsf{N}-1}{\ell-1}$ ℓ-blocks.[2] By using the MAN cache placement to store each ℓ-block at the user sides, [21] proposed a delivery phase which contains N steps. In Step ℓ, only ℓ-blocks are transmitted. In addition, there are $\binom{\mathsf{N}-1}{\ell-1}$ rounds for the transmission of Step ℓ, and each round is treated as an MAN caching problem with K users, each of which should decode exactly one ℓ-block. Then the caching scheme in [4] was used to transmit packets for each round. The caching schemes in [19] and [21], were extended in [30] and [31] to caching problems with correlated files where the correlation is dynamic and the channel is a Gaussian broadcast channel, respectively.

2) Coded Caching With Multiple Requests: Each step of the caching problem with correlated files of combinatorial overlaps in [21] is a special case of coded caching with multiple requests originally proposed in [32] and [33], where the library contains N equal-length and independent files and each user demands L files from the library.[3] By using the MAN placement and an index coding delivery, the achieved worst-case load in [32] is order optimal to within a factor of 18, while the achieved average load over the uniform demand distribution in [33] is order optimal to within a factor of 12 when the numbers of files and users go to infinity. With the MAN placement, a multi-round delivery scheme was proposed in [34], where the delivery phase is divided into L rounds and in each round the MAN delivery scheme in [2] is used to let each user decode one file. The worst-case load of this multi-round scheme was proved to be order optimal to within a factor of 11 [34].

Instead of using the MAN scheme in each round, the authors in [35] proposed to use the caching scheme in [4] to leverage the multicast opportunities. In addition, by considering all the L rounds, an overall transmission coding matrix can be generated. If the coding matrix is not full-rank, the caching scheme in [35] then takes the full-rank sub-matrix. This delivery scheme was proved to be optimal under the constraint of the MAN placement for demands with $\mathsf{K} \leq 4$, $\mathsf{M} = \mathsf{N}/\mathsf{K}$, and $\mathsf{L} = 2$, with the exception of one demand for $\mathsf{K} = 3$ and three demands for $\mathsf{K} = 4$.

Coded caching with multiple requests where the users demand different numbers of files, was considered in [36], [37], [38], [39], [40], [41], [42], [43], and [44].[4] The caching schemes in [36], [38], [39], [40], and [37] are based on the round-division strategy as described above while the schemes in [41], [42] use coded cache placements for some small memory size regimes and the schemes in [43] and [44] are based on a combinatorial structure referred to as placement delivery array (PDA) originally proposed in [45].

Most of the existing works divide the multi-request problem into a sequence of single-request problems (as in [21], [34], [35], [36], [37], [38], [39], [40]). There are three main limitations in dividing the delivery into single-request problems, namely (1) the same file may exist in different rounds and this round-division method may miss some multicast opportunities, (2) even if there does not exist file overlap cross different rounds, this round-division method still cannot fully leverage the multicast opportunities (as illustrated in Example V-A),

[1] Common information between correlated files may be defined in different contexts, such as Gács–Körner–Witsenhausen common information, mutual information, Wyner's common information. These three quantities coincide if common information between some files is represented by the overlaps of these files [18, Section 14.2]. In addition, as stated in [19], for any file correlation model, by compressing the library using the Gray-Wyner network [20], the resulting description of the library reduces to an overlap model, where each subset of compressed files contains a common part.

[2] While obviously an idealization, this symmetric combinatorial overlap model can capture several interesting scenarios. For example, suppose that each file is a collection of subfiles, e.g., a photo album, and some photos are in common between different albums (same can be said for playlists, where some songs are common to different playlists). Note that in information theory, ideally symmetric combinatorial models are considered in several problems in order to get clean theoretic results. For example, in the literature of coded caching, symmetric combinatorial models are widely considered to make the problems theoretically tractable, such as two-hop combination networks with relays [9], [10], [22], [23], [24], [25], combinatorial combinatorial multi-access networks [26], [27], coded caching with combinatorial file demand sets [28], Map-Reduce coded distributed computation models [29], etc. In addition, lack of perfect symmetry in the combinatorial file overlap topology, converse bounds should be derived case-by-case and the proposed achievable scheme for the symmetric case can be still used by adding some virtual common parts.

[3] This is because in [21] the ℓ-blocks in Step ℓ are assumed to be independent and to have the same length. Thus we can treat each block as one independent file in the coded caching problem with multiple requests, while each user requests $\binom{\mathsf{N}-1}{\ell-1}$ blocks.

[4] The coded caching problem with shared caches was considered in [38], [39], [40], [41], [42], [43], and [44], where the model contains a central server and U cache-nodes. Each cache-node i is connected to L_i cache-less users, where each user can access to only one cache-node. This problem can be seen as a special case of the coded caching problem with multiple requests, where each user i demands L_i files.

Authorized licensed use limited to: University of Illinois at Chicago Library. Downloaded on November 08,2023 at 23:24:46 UTC from IEEE Xplore. Restrictions apply.

and (3) finding the best division of the users' demands into L groups is computationally hard.

B. Contributions

If one directly considers the most general problem of correlated files, it is very challenging to make general optimality statements. In this paper, as in [21] we consider a symmetric combinatorial version of the problem; in addition, the considered correlation model is the following simplification of the correlation model in [21]: we fix one $\mathsf{r} \in [\mathsf{N}]$ and assume each file only contains r-blocks,[5] for which we propose a new interference alignment based delivery scheme, which jointly serves the users' demands instead of dividing the delivery into single-request problems. Our main contributions are as follows:

1) We derive a converse bound on the minimal average load over the uniform demand distribution under the constraint of uncoded cache placement, by leveraging the acyclic index coding converse bound in [46].
2) By jointly serving the users' demands, we propose a caching scheme for the case where every user demands a distinct file. The achieved load matches our proposed converse bound under the constraint of uncoded cache placement.
3) By combining the above achievable scheme with an interference alignment idea, we then propose a delivery scheme for general demands containing two sub-phases, where the first sub-phase is the same as the one for distinct demand case and the additional second sub-phase is used to align interference at the various users. The proposed caching scheme is proved to be order optimal to within a factor of 2 in terms of the average load over the uniform demand distribution.
4) By further cancelling interference, we prove that the second sub-phase in the delivery is not necessary, thus resulting in the exact optimal average load under the constraint of uncoded cache placement and uniform demand distribution, for the following two cases: (i) $\mathsf{KrM} \leq 2\mathsf{N}$ or $\mathsf{KrM} \geq (\mathsf{K}-1)\mathsf{N}$ or $\mathsf{r} \in \{1, 2, \mathsf{N}-1, \mathsf{N}\}$, and (ii) the number of distinct demanded files is no larger than four.
5) As an extension, the above exact and order optimal results can be extended to the worst-case loads.
6) As a by-product, an extension of the proposed scheme for $\mathsf{M} = \mathsf{N}/\mathsf{K}$ is optimal under the constraint of MAN placement for the four cases left open in [35] of the coded caching problem with multiple requests. As another by-product, the proposed scheme for distinct demands can be extended to the coded distributed computing problem with a central server, which achieves the optimal transmission load over the binary field.

C. Paper Organization

The rest of the paper is organized as follows. The system model for the considered coded caching problem with correlated files of combinatorial overlaps is given in Section II. In Section III, our main results and some numerical evaluations are presented. The proofs of the proposed converse bound and achievable schemes are given in Sections IV and V, respectively. Section VI concludes the paper. The proofs of some auxiliary results can be found in Appendices.

[5] Clearly, the proposed achievable schemes could be directly applied into the correlation model in [21].

D. Notation Convention

Calligraphic symbols denote sets, bold symbols denote vectors, and sans-serif symbols denote system parameters. We use $|\cdot|$ to represent the cardinality of a set or the length of a vector; $[a:b] := \{a, a+1, \ldots, b\}$ and $[n] := [1, 2, \ldots, n]$; $\oplus$ represents bit-wise XOR. We let $\binom{x}{y} = 0$ if $x < 0$ or $y < 0$ or $x < y$.

II. System Model

In an $(\mathsf{N}, \mathsf{K}, \mathsf{r}, \mathsf{M})$ shared-link caching problem with correlated files of combinatorial overlaps, a server has access to a library of $\mathsf{N} \in \mathbb{N}$ files (each of which contains $\mathsf{B} \in \mathbb{N}$ iid bits) denoted by $F_1, \ldots, F_{\mathsf{N}}$. The server is connected to $\mathsf{K} \in \mathbb{N}$ users through a shared error-free link. Each file is composed of $\binom{\mathsf{N}-1}{\mathsf{r}-1}$ independent and equal-length blocks, where $\mathsf{r} \in [\mathsf{N}]$; we denote

$$F_i = \{W_{\mathcal{S}} : \mathcal{S} \subseteq [\mathsf{N}], |\mathcal{S}| = \mathsf{r}, i \in \mathcal{S}\}, \ \forall i \in [\mathsf{N}], \tag{1}$$

where the block $W_{\mathcal{S}}$ represents the exclusive common part across the files indexed by $\mathcal{S}$. Hence, in the whole library there are $\binom{\mathsf{N}}{\mathsf{r}}$ independent blocks, each of which has $\mathsf{B}/\binom{\mathsf{N}-1}{\mathsf{r}-1}$ bits. A coded caching scheme has two phases: placement and delivery.

A. Placement Phase

During the cache placement phase, user $k \in [\mathsf{K}]$ stores information about the N files in its cache of size MB bits, where $\mathsf{M} \in [0, \mathsf{N}/\mathsf{r}]$. This phase is done without knowledge of the users' demands. We denote the content in the cache of user $k \in [\mathsf{K}]$ by Z_k, where

$$H(Z_k | F_1, \ldots, F_{\mathsf{N}}) = 0, \ \forall k \in [\mathsf{K}]. \tag{2}$$

We let $\mathbf{Z} := (Z_1, \ldots, Z_{\mathsf{K}})$.

B. Delivery Phase

During the delivery phase, user $k \in [\mathsf{K}]$ demands file F_{d_k} where $d_k \in [\mathsf{N}]$. The demand vector $\mathbf{d} := (d_1, \ldots, d_{\mathsf{K}})$ is revealed to all nodes. Given $(\mathbf{d}, \mathbf{Z})$, the server broadcasts a message $X(\mathbf{d}, \mathbf{Z})$ of $\mathsf{B} \cdot \mathsf{R}(\mathbf{d}, \mathbf{Z})$ bits to all users, where

$$H(X(\mathbf{d}, \mathbf{Z}) | \mathbf{d}, F_1, \ldots, F_{\mathsf{N}}) = 0, \ \forall \mathbf{d} \in [\mathsf{N}]^{\mathsf{K}}. \tag{3}$$

User $k \in [\mathsf{K}]$ must recover its desired file F_{d_k} from Z_k and $X(\mathbf{d}, \mathbf{Z})$, where

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

Authorized licensed use limited to: University of Illinois at Chicago Library. Downloaded on November 08,2023 at 23:24:46 UTC from IEEE Xplore. Restrictions apply.

C. Load

For each demand vector $\mathbf{d}$, we define $\mathcal{N}_{\mathbf{d}}(\mathcal{T}) := \{d_k : k \in \mathcal{T}\}$ as the set of demanded files by users in $\mathcal{T}$, where $\mathcal{T} \subseteq [\mathsf{K}]$. A demand vector $\mathbf{d}$ is said to be of type $\mathcal{D}_{N_{\mathrm{e}}(\mathbf{d})}$ if it has $N_{\mathrm{e}}(\mathbf{d}) := |\mathcal{N}_{\mathbf{d}}([\mathsf{K}])|$ distinct entries. Based on the uniform demand distribution, the objective is to determine the optimal average load among all demands of the same type; that is

$$\mathsf{R}^{\star}(\mathsf{M}, s) := \min_{\mathbf{Z}} \ \mathbb{E}_{\mathbf{d}\in\mathcal{D}_s}\left[\min_{X(\mathbf{d},\mathbf{Z})} \mathsf{R}(\mathbf{d},\mathbf{Z})\right], \tag{5}$$

for all $s \in [\min\{\mathsf{K},\mathsf{N}\}]$, and the optimal average load among all possible demands; that is

$$\mathsf{R}^{\star}(\mathsf{M}) := \min_{\mathbf{Z}} \ \mathbb{E}_{\mathbf{d}\in[\mathsf{N}]^{\mathsf{K}}}\left[\min_{X(\mathbf{d},\mathbf{Z})} \mathsf{R}(\mathbf{d},\mathbf{Z})\right]. \tag{6}$$

Note that, $\mathsf{R}^{\star}(\mathsf{M}) \neq \mathbb{E}_s[\mathsf{R}^{\star}(\mathsf{M}, s)]$ in general, unless the same cache placement policy optimizes the load in (5) for all $s \in [\min\{\mathsf{K},\mathsf{N}\}]$. In addition, in this paper when we discuss the average load, we only consider the uniform demand distribution; thus for the sake of conciseness, we will not specify the demand distribution in the rest of the paper.

In addition, we also define the optimal worst-case load over all possible demands as

$$\mathsf{R}^{\star}_{\mathrm{worst}}(\mathsf{M}) := \min_{\mathbf{Z}} \max_{\mathbf{d}\in[\mathsf{N}]^{\mathsf{K}}} \min_{X(\mathbf{d},\mathbf{Z})} \mathsf{R}(\mathbf{d},\mathbf{Z}). \tag{7}$$

D. Uncoded Cache Placement

The cache placement policy is said to be *uncoded* if each user directly copies some library bits into its cache. The optimal loads under the constraint of uncoded cache placement are denoted by $\mathsf{R}^{\star}_{\mathrm{u}}(\mathsf{M}, s)$, $\mathsf{R}^{\star}_{\mathrm{u}}(\mathsf{M})$, and $\mathsf{R}^{\star}_{\mathrm{u,worst}}(\mathsf{M})$ are defined similarly to (5), (6), and (7), respectively.

Note that, in this paper we mainly focus on the average loads and then extend the obtained results to the worst case.

Remark 1 (Special Cases): Our model reduces to the MAN coded caching problem when $\mathsf{r} = 1$, and to the case of a library with a single file when $\mathsf{r} = \mathsf{N}$. Both cases are either solved exactly or to within a factor of 2 in [5]. □

Remark 2 (Relation to the More General Coded Caching Problem with Correlated Files of Combinatorial Overlaps): In this paper, in order to make fundamental progress on the problem of caching correlated content, we simplify the model [21] as follows. In [21] a certain parameter ℓ ranges from 1 to the number of files in the system (each ℓ_1 files have a common part, each ℓ_2 files also have a common part, etc.), while in our model ℓ is fixed to a single value r. Our model is thus a special case of the one in [21]. Using our models however, we can make conclusive statements (either exact capacity results, or capacity to within a constant multiplicative gap) which were not in [21]. □

Remark 3 (Relation to the Coded Caching Problem with Multiple Requests): If we identify the $\binom{\mathsf{N}}{\mathsf{r}}$ independent blocks as files of a library, and allow each cache-equipped user to request $\binom{\mathsf{N}-1}{\mathsf{r}-1}$ such blocks/files, the considered caching problem with correlated files of combinatorial overlaps relates to the *symmetric* coded caching problem with multiple requests considered in [32], where 'symmetric' means that each user requests the same number of files, which is equal to $\binom{\mathsf{N}-1}{\mathsf{r}-1}$. However, there is a strong structure of the users' demands in our problem, while in [32] each user can demand arbitrary L files.

The relationship among the two problems can be also explained as follows. For the case of multiple requests, assume that the $\binom{\mathsf{N}}{\mathsf{r}}$ independent files are equally popular. On average, each of such independent files will appear on average the same number of times over the ensemble of all possible multiple request configurations. We construct N such multiple request configurations, each of which is formed by $\binom{\mathsf{N}-1}{\mathsf{r}-1}$ independent files (in fact, each multiple request configuration corresponds to a 'file' in the correlated file library of our problem). It follows that each independent file appears on average $\mathsf{N}\binom{\mathsf{N}-1}{\mathsf{r}-1}/\binom{\mathsf{N}}{\mathsf{r}} = \mathsf{r}$ times in the ensemble of possible multiple requests configurations. If instead of random multiple requests, we consider the deterministic symmetric case, where the possible multiple requests configurations are all and only those for which each independent files appears exactly r times (and not on average r times), we have the exact equivalence of our problem with the case of multiple requests of independent files. With this interpretation, the proposed results in this paper also shed light into the very relevant and intricate problem of how to handle optimally the case where each user makes a sequence of requests of independent files (blocks). The fact that there are repeated elements in such sequence of requests is a 'fundamental' aspect of caching (also in practice), where one needs to devise schemes that take advantage of previous requests and do not send the same coded bits multiple times. □

III. Main Results and Numerical Evaluations

In this section, we state our main results and present numerical evaluations of the proposed converse and achievable bounds. We shall use the subscripts 'u,conv' and 'u,ach' for converse (conv) and achievable (ach) bounds, respectively, under the constraint of uncoded cache placement (u).

A. Converse Bound

Inspired by [3], we use the acyclic index coding converse bound from [46] to derive the following converse bound under the constraint of uncoded cache placement for our problem. The proof can be found in Section IV.

Theorem 1 (Converse): For an $(\mathsf{N},\mathsf{K},\mathsf{r},\mathsf{M})$ shared-link caching problem with correlated files of combinatorial overlaps, $\mathsf{R}^{\star}_{\mathrm{u}}(\mathsf{M}, s)$, $s \in [\min\{\mathsf{K},\mathsf{N}\}]$, is lower bounded by the lower convex envelope of the following (M,R) pairs

$$\left(\frac{\mathsf{N}t}{\mathsf{K}\mathsf{r}}, c_t^s\right)_{\mathrm{u,conv}}, \ \forall t \in [0:\mathsf{K}], \tag{8}$$

where

$$c_t^s := \frac{\sum_{j\in[\min\{s,\mathsf{N}-\mathsf{r}+1,\mathsf{K}-t\}]}\binom{\mathsf{N}-j}{\mathsf{r}-1}\binom{\mathsf{K}-j}{t}}{\binom{\mathsf{N}-1}{\mathsf{r}-1}\binom{\mathsf{K}}{t}}. \tag{9}$$

Authorized licensed use limited to: University of Illinois at Chicago Library. Downloaded on November 08,2023 at 23:24:46 UTC from IEEE Xplore. Restrictions apply.

In addition, $\mathsf{R}_{\mathrm{u}}^{\star}(\mathsf{M})$ is lower bounded by the lower convex envelope of the following (M, R) pairs

$$\left(\frac{\mathsf{N}t}{\mathsf{K}\mathsf{r}}, \mathbb{E}_{\mathbf{d}\in[\mathsf{N}]^{\mathsf{K}}}\left[c_t^{N_{\mathrm{e}}(\mathbf{d})}\right]\right)_{\mathrm{u,conv}}, \ \forall t \in [0:\mathsf{K}]. \tag{10}$$

□

Theorem 1 for $\mathsf{r}=1$ recovers the converse result for the MAN scheme under uncoded placement in [4]; in particular the worst-case load is obtained for $s=\min\{\mathsf{K},\mathsf{N}\}$ in (8), while the average load is given by (10). Theorem 1 for $\mathsf{r}=\mathsf{N}$ recovers the converse result for the MAN scheme with a single file in the library; that is, $c_t^s = 1-t/\mathsf{K} \iff \mathsf{R}^{\star}(\mathsf{M}) = 1-\mathsf{M}$ for $\mathsf{M}\in[0,1]$.

B. Achievable Schemes

Let $\mathsf{M}=\frac{\mathsf{N}t}{\mathsf{K}\mathsf{r}}$ for some integer $t\in[0:\mathsf{K}]$. Recall that we denote by $N_{\mathrm{e}}(\mathbf{d})$ the number of distinct files in the demand vector $\mathbf{d}$. For each demanded file, we pick a leader user demanding this file. The set of chosen leader users for the demand vector $\mathbf{d}$ is denoted by $\mathcal{L}(\mathbf{d}) = \{u_1,\ldots,u_{N_{\mathrm{e}}(\mathbf{d})}\}$. For each subset of users $\mathcal{T}\subseteq[\mathsf{K}]$, the set of leader users demanding the files $\mathcal{N}_{\mathbf{d}}(\mathcal{T})$ is denoted by $\mathcal{L}_{\mathbf{d}}(\mathcal{T})$.

We propose the following achievable scheme, which is analyzed in Section V.

Block subdivision: $\forall \mathcal{S}\subseteq[\mathsf{N}]:|\mathcal{S}|=\mathsf{r}$, let

$$W_{\mathcal{S}} = \{W_{\mathcal{S},\mathcal{V}} : \mathcal{V}\subseteq[\mathsf{K}], |\mathcal{V}|=t\}. \tag{11a}$$

Placement Phase: $\forall k\in[\mathsf{K}]$, let

$$Z_k = \{W_{\mathcal{S},\mathcal{V}} : \mathcal{S}\subseteq[\mathsf{N}], |\mathcal{S}|=\mathsf{r}, \mathcal{V}\subseteq[\mathsf{K}], |\mathcal{V}|=t, k\in\mathcal{V}\}. \tag{11b}$$

Delivery sub-phase 1:

$$\forall j\in[\min\{N_{\mathrm{e}}(\mathbf{d}), \mathsf{N}-\mathsf{r}+1, \mathsf{K}-t\}], \tag{11c}$$

$$\forall \mathcal{J}\subseteq[\mathsf{K}]\setminus\{u_1,\ldots,u_{j-1}\} : |\mathcal{J}|=t+1, u_j\in\mathcal{J}, \tag{11d}$$

$$\forall \mathcal{B}\subseteq[\mathsf{N}]\setminus\{d_{u_1},\ldots,d_{u_j}\} : |\mathcal{B}|=\mathsf{r}-1, \tag{11e}$$

send a multicast message $C_{\mathcal{J},\mathcal{B}}$ as defined in (28). (11f)

Delivery sub-phase 2:

$$\forall j\in[\min\{N_{\mathrm{e}}(\mathbf{d}), \mathsf{N}-\mathsf{r}+1, \mathsf{K}-t\}], \tag{11g}$$

$$\forall q\in[j+1:\min\{\mathsf{N}-\mathsf{r}+2, \mathsf{K}-t+1, N_{\mathrm{e}}(\mathbf{d})\}], \tag{11h}$$

$$\forall \mathcal{J}\subseteq[\mathsf{K}]\setminus\{u_1,\ldots,u_{q-1}\}\cup\{u_j\} : |\mathcal{J}|=t+1, \{u_j,u_q\}\subseteq\mathcal{J}, \mathcal{J}\cap\{u_{q+1},\ldots,u_{N_{\mathrm{e}}(\mathbf{d})}\}\neq\emptyset, \tag{11i}$$

$$\forall \mathcal{B}\subseteq[\mathsf{N}]\setminus\{d_{u_1},\ldots,d_{u_q}\} : |\mathcal{B}|=\mathsf{r}-2, \mathcal{B}\cap\mathcal{N}_{\mathbf{d}}([\mathsf{K}])\neq\emptyset, \tag{11j}$$

send a multicast message $C_{\mathcal{J},\mathcal{B}}$ as defined in (28). (11k)

In the rest of this section we analyze the above proposed scheme in various settings with an increasing order of complexity. Since the scheme is highly combinatorial, we shall start with a case that is the simplest to analyze and that brings to bear some of the key ideas. We shall then show that the a similar analysis applies also to more complex scenarios. In the following, optimality is understood under the constraint of uncoded cache placement. In particular, we develop these concepts in the following order:

- In Section III-C we show that the general scheme in (11) with only the first delivery sub-phase allows each leader user to decode its desired file. We also show that the first sub-phase alone is exactly optimal when the users request different files; that is, all users are leaders, which is possible when $N_{\mathrm{e}}(\mathbf{d})=\mathsf{K}\leq\mathsf{N}$.
- In Section III-D we show that the scheme in (11), with both delivery sub-phases, can satisfy every user regardless of the demand type, where the transmissions in sub-phase 2 are used to cancel the interferences experienced by the non-leader users. We also show its optimality to within a factor of 2 for any demand type.
- In Section III-E we show that for some cases (such as, for example, the case of either small or large memory size), each non-leader can reconstruct its required transmitted multicast messages in sub-phase 2 by performing linear combinations of the transmitted multicast messages in sub-phase 1; that is, sub-phase 2 is redundant. For these cases, we show the exact optimality.
- In Section III-F we show how the scheme in (11) can be used for other caching problems, by either offering simpler codes for the delivery phase than those known in the literature, or by providing an optimal scheme outperforming state-of-the-art schemes.
- In Section III-G we present some numerical evaluations of the proposed bounds.

C. Optimality of (11) for Demand Type $\mathcal{D}_s$ Where $s=\mathsf{K}\leq\mathsf{N}$

We consider the case where each user makes a distinct request, which requires $\mathsf{K}\leq\mathsf{N}$ and demand type $\mathcal{D}_s$ with $s=\mathsf{K}=N_{\mathrm{e}}(\mathbf{d})$. We propose a caching scheme where we *jointly serve the users' demands*. Existing methods approach the problem by serving requests in multiple rounds [21], [34], [35], where each round is a single-request MAN scheme. Our scheme is as in (11), but where only the first sub-phase of the delivery phase takes place. In particular, for $\mathsf{M}=\frac{\mathsf{N}t}{\mathsf{K}\mathsf{r}}$ and $s=N_{\mathrm{e}}(\mathbf{d})=\mathsf{K}\leq\mathsf{N}$, our proposed delivery phase contains $\min\{\mathsf{N}-\mathsf{r}+1,\mathsf{K}-t\}$ steps, where in each step we transmit multicast messages to satisfy one leader user at a time. After all steps are done, the remaining $\mathsf{K}-\min\{\mathsf{N}-\mathsf{r}+1,\mathsf{K}-t\}$ users (who are also leaders, since here we consider a distinct request for each user) can also recover their desired files. The achieved load is presented in the following theorem, whose proof can be found in Section V-B.

Theorem 2 (Optimality for Distinct Requests): For an $(\mathsf{N},\mathsf{K},\mathsf{r},\mathsf{M})$ shared-link caching problem with correlated files of combinatorial overlaps where $\mathsf{N}\geq\mathsf{K}$, $\mathsf{R}_{\mathrm{u}}^{\star}(\mathsf{M},\mathsf{K})$ is the lower convex envelop of the points $\left(\frac{\mathsf{N}t}{\mathsf{K}\mathsf{r}}, c_t^{\mathsf{K}}\right)_{\mathrm{u,conv}}$ where $t\in[0:\mathsf{K}]$ and c_t^{K} is defined in (9), which is achieved by the scheme in (11) with only the first delivery sub-phase. □

D. Performance of (11) for Any Demand Type

We analyze here the scheme in (11) with two sub-phases in the delivery phase, and show that it is able to satisfy

Authorized licensed use limited to: University of Illinois at Chicago Library. Downloaded on November 08,2023 at 23:24:46 UTC from IEEE Xplore. Restrictions apply.

general demands. The main ingredients of the scheme are as follows. In the first delivery sub-phase, we generate multicast messages in (11f) so that each leader user can recover its desired file by the end of this sub-phase; in the second delivery sub-phase, we transmit some additional multicast messages in (11k), so that each non-leader user can cancel all non-intended (aligned interference) sub-blocks from the received multicast messages useful to it and thus can eventually recover its desired file. The achieved load is presented in the following theorem, whose proof can be found in Section V-C.

Theorem 3 (Interference-Alignment Based Delivery Scheme): For an $(\mathsf{N},\mathsf{K},\mathsf{r},\mathsf{M})$ shared-link caching problem with correlated files of combinatorial overlaps, for any $s \in [\min\{\mathsf{K},\mathsf{N}\}]$, $\mathsf{R}^{\star}_{\mathrm{u}}(\mathsf{M},\mathsf{s})$ is upper bounded by the by the lower convex envelope of the following (M,R) pairs

$$\left(\frac{\mathsf{N}t}{\mathsf{Kr}}, c_t^s + e_t^s\right)_{\text{u,ach}}, \ \forall t \in [0:\mathsf{K}], \tag{12}$$

where c_t^s is defined in (9) and e_t^s is defined in (13), shown at the bottom of the next page.

In addition, $\mathsf{R}^{\star}_{\mathrm{u}}(\mathsf{M})$ is upper bounded by the lower convex envelope of the following (M,R) pairs

$$\left(\frac{\mathsf{N}t}{\mathsf{Kr}}, \mathbb{E}_{\mathbf{d}\in[\mathsf{N}]^{\mathsf{K}}}\left[c_t^{N_{\mathrm{e}}(\mathbf{d})} + e_t^{N_{\mathrm{e}}(\mathbf{d})}\right]\right)_{\text{u,ach}}, \ \forall t \in [0:\mathsf{K}]. \tag{14}$$

□

By comparing the converse bound in Theorem 1 and the achievable bound in Theorem 3, we have the following result, whose proof can be found in Section V-E.

Theorem 4 (Order Optimality for Theorem 3): For an $(\mathsf{N},\mathsf{K},\mathsf{r},\mathsf{M})$ shared-link caching problem with correlated files of combinatorial overlaps, under the constraint of uncoded cache placement, the achieved average loads in (12) and (14) are order optimal to within a factor of 2, for any demand type $\mathcal{D}_s$ where $s \in [\min\{\mathsf{K},\mathsf{N}\}]$ and all possible demands, respectively. □

E. Optimality of (11) for $\mathsf{r} \in \{1, 2, \mathsf{N}-1, \mathsf{N}\}$ or $t \in \{0, 1, 2, \mathsf{K}-1, \mathsf{K}\}$ or $s \in [\min\{\mathsf{N},\mathsf{K},4\}]$

In Theorem 3, c_t^s in (9) is the load for the first delivery sub-phase while e_t^s in (13) is the load for the second delivery sub-phase. Hence, compared to the converse bound in Theorem 1, e_t^s is the term leading to the sub-optimality. In Theorem 2, where we showed the exact optimality for distinct demands, the second sub-phase is not needed. We investigate here the other cases where the second sub-phase is not needed. *In these cases, each non-leader user can reconstruct its required multicast messages sent in sub-phase 2 by linearly combining multicast messages sent in sub-phase 1.* Thus we obtain the following exact optimality result proved in Section V-F.

Theorem 5 (Exact Optimality for Some Cases): For an $(\mathsf{N},\mathsf{K},\mathsf{r},\mathsf{M})$ shared-link caching problem with correlated files of combinatorial overlaps, we have that $\mathsf{R}^{\star}_{\mathrm{u}}(\mathsf{M},s)$ and $\mathsf{R}^{\star}_{\mathrm{u}}(\mathsf{M})$ are equal to the lower convex envelops of $\left(\frac{\mathsf{N}t}{\mathsf{Kr}}, c_t^s\right)$ and of $\left(\frac{\mathsf{N}t}{\mathsf{Kr}}, \mathbb{E}_{\mathbf{d}\in[\mathsf{N}]^{\mathsf{K}}}\left[c_t^{N_{\mathrm{e}}(\mathbf{d})}\right]\right)$, respectively, where c_t^s is defined in (9), in the following cases:

1) Case 1 (small or large file correlation): when $\mathsf{r} \in \{1, 2, \mathsf{N}-1, \mathsf{N}\}$, where the optimality holds for any $s \in [\min\{\mathsf{K},\mathsf{N}\}]$ and any $t \in [0:\mathsf{K}]$;
2) Case 2 (small or large cache size): when $t \in \{0, 1, 2, \mathsf{K}-1, \mathsf{K}\}$, where the optimality holds for any $s \in [\min\{\mathsf{K},\mathsf{N}\}]$ and any $\mathsf{r} \in [\mathsf{N}]$;
3) Case 3 (small number of distinct requests): when $s \in [\min\{\mathsf{K},\mathsf{N},4\}]$, where the optimality holds for any $\mathsf{r} \in [\mathsf{N}]$ and any $t \in [0:\mathsf{K}]$. In this case, no claim can be made on $\mathsf{R}^{\star}_{\mathrm{u}}(\mathsf{M})$ as only some values of s are exactly characterized.

□

From Theorem 5 we immediately have the following corollary, which can be proved straightforwardly by noting that Theorem 5.Case 3 covers all possible values of s when $\min(\mathsf{N},\mathsf{K}) \le 4$.

Corollary 1: For an $(\mathsf{N},\mathsf{K},\mathsf{r},\mathsf{M})$ shared-link caching problem with correlated files of combinatorial overlaps, the converse bounds in (8) and (10) are achievable when $\min(\mathsf{N},\mathsf{K}) \le 4$ by the scheme in (11) with only the first delivery sub-phase.

□

As a result of Theorems 2, 3, and 5, the best achievable bound by the proposed schemes is the lower convex envelop of the following (M,R) pairs

$$\left(\frac{\mathsf{N}t}{\mathsf{Kr}}, c_t^s + e_t^s \cdot \mathbb{1}_{s\in[5:\mathsf{K}-1],t\in[3:\mathsf{K}-2],\mathsf{r}\in[3:\mathsf{N}-2]}\right)_{\text{u,ach}}, \forall t \in [0:\mathsf{K}]. \tag{15}$$

Note that where $\mathbb{1}$ is the indicator function: $\mathbb{1}_{\text{event}} = 1$ if event is true and $\mathbb{1}_{\text{event}} = 0$ otherwise.

F. Extensions

Our results can be used in models other than the one considered in this paper. Examples are as follows.

1) Extension to the Worst-Case Load: The proposed achievability, converse, and optimality results can be also extended to the case of worst-case load, since the optimal worst-case load under uncoded cache placement is also lower bounded by (8) for any $s \in [\min\{\mathsf{N},\mathsf{K}\}]$ and c_t^s increases with s.

Corollary 2: For an $(\mathsf{N},\mathsf{K},\mathsf{r},\mathsf{M})$ shared-link caching problem with correlated files of combinatorial overlaps, under the constraint of uncoded cache placement, the optimal worst-case load over all possible demands under uncoded cache placement $\mathsf{R}^{\star}_{\mathrm{u,worst}}(\mathsf{M})$ is upper bounded by the lower convex envelope of the following (M,R) pairs

$$\left(\frac{\mathsf{N}t}{\mathsf{Kr}}, \max_{s\in[\min\{\mathsf{N},\mathsf{K}\}]} c_t^s + e_t^s \cdot \mathbb{1}_{s\in[5:\mathsf{K}-1],t\in[3:\mathsf{K}-2],\mathsf{r}\in[3:\mathsf{N}-2]}\right), \tag{16}$$

and lower bounded by the lower convex envelope of the following (M,R) pairs

$$\left(\frac{\mathsf{N}t}{\mathsf{Kr}}, c_t^{\min\{\mathsf{N},\mathsf{K}\}}\right). \tag{17}$$

The achieved worst-case load in (16) is order optimal to within a factor of 2 under the constraint of uncoded cache placement. In addition, the achieved worst-case load in (16)

Authorized licensed use limited to: University of Illinois at Chicago Library. Downloaded on November 08,2023 at 23:24:46 UTC from IEEE Xplore. Restrictions apply.

is optimal under uncoded cache placement in the following cases: Theorem 5.Case 1, Theorem 5.Case 2, and Corollary 1. □

Remark 4 (Average and Worst-Case Loads): Most past works only aimed to design schemes that minimize the worst-case load, such as those in [21] (for caching with correlated files of combinatorial overlaps) and in [32] and [34] (for caching with multiple requests). Order optimality results (to within factors 11 and 18) on the worst-case load were derived in [32] and [34] for caching with multiple requests. The order optimality on the average load over the uniform demand distribution for caching with multiple requests was characterized to within a factor of 12 when the numbers of files and users go to infinity [33]. To the best of our knowledge, (except the cases of two files and three files) no specific order optimality results are known specifically for caching with correlated files of combinatorial overlaps. Therefore, a major contribution of this paper, besides sharpening existing results for caching with multiple requests, is to have derived (exact or order) optimality results for caching with correlated files of combinatorial overlaps for *any demand type* or *over all possible demands*, under the constraint of uncoded cache placement. □

2) Extension to the More General Coded Caching Problem With Correlated Files: As already mentioned earlier, in this paper we simplify the model [21] by fixing the parameter ℓ in [21] to be equal to r (as opposed to let it be in a range). We can extend our results to the case where ℓ is in a range as follows. If ℓ is in a range as considered in [21], we can construct a caching scheme by 'memory-sharing' among the proposed scheme in Theorems 2, 3, and 5 as follows.

- Library. Assume that the length of each block $W_{\mathcal{S}}$, where $\mathcal{S} \subseteq [\mathsf{N}]$ and $|\mathcal{S}| = \ell$, is $\mathsf{p}_\ell \mathsf{B}/\binom{\mathsf{N}}{\ell}$. Let $\mathsf{p}_\ell \in [0,1]$ and $\sum_{\ell \in [\mathsf{N}]} \mathsf{p}_\ell = 1$. The values $(\mathsf{p}_\ell : \ell \in [\mathsf{N}])$ are assumed to be fixed system parameters.
- Placement. Choose integers $t_\ell \in [\mathsf{K}]$ for $\ell \in [\mathsf{N}]$, We partition block $W_{\mathcal{S}}$ into $\binom{\mathsf{K}}{t_{|\mathcal{S}|}}$ equal-length sub-blocks and denote $W_{\mathcal{S}} = \{W_{\mathcal{S},\mathcal{V}} : \mathcal{V} \subseteq [\mathsf{K}], |\mathcal{V}| = t_{|\mathcal{S}|}\}$. User $k \in [\mathsf{K}]$ caches sub-block $W_{\mathcal{S},\mathcal{V}}$ if $k \in \mathcal{V}$, which requires a cache of size
$$\mathsf{M} = \sum_{\ell \in [\mathsf{N}]} \frac{\mathsf{N} t_\ell \mathsf{p}_\ell}{\mathsf{K}\ell}. \tag{18}$$
- Delivery. For demand vector $\mathbf{d}$, if $N_{\mathrm{e}}(\mathbf{d}) = \mathsf{K}$ (i.e., each user demands a distinct file) or $N_{\mathrm{e}}(\mathbf{d}) \in [\min\{\mathsf{K}, \mathsf{N}, 4\}]$, we use the proposed caching scheme only with the first delivery sub-phase and the achieved load is
$$\mathsf{R} = \sum_{\ell \in [\mathsf{N}]} \mathsf{p}_\ell c^{\mathsf{K}}_{t_\ell}. \tag{19}$$
If $\min\{\mathsf{K}, \mathsf{N}, 4\} < s = N_{\mathrm{e}}(\mathbf{d}) < \mathsf{K}$, we have two cases: (i) if either $\ell \in \{1, 2, \mathsf{N}-1, \mathsf{N}\}$ or $t_\ell \in \{0, 1, 2, \mathsf{K}-1, \mathsf{K}\}$, we use the proposed caching scheme only with the first delivery sub-phase to encode all blocks; (ii) otherwise, we use the proposed caching scheme with two delivery sub-phases. Hence, the achieved load is
$$\mathsf{R} = \sum_{\ell_1 \in \{1,2,\mathsf{N}-1,\mathsf{N}\}} \mathsf{p}_{\ell_1} c^s_{t_{\ell_1}} + \sum_{\ell_2 \in [3:\mathsf{N}-2]} \left(\mathsf{p}_{\ell_2} c^s_{t_{\ell_2}} + \mathsf{p}_{\ell_2} e^s_{t_{\ell_2}} \mathbb{1}_{t_{\ell_2} \notin \{0,1,2,\mathsf{K}-1,\mathsf{K}\}} \right). \tag{20}$$
- The achievable memory-load tradeoff is the lower convex envelope of the above points for all possible $\mathbf{t} := (t_\ell : \ell \in [\mathsf{N}])$.

3) Extension to the Coded Caching Problem With Multiple Requests: Our proposed strategy which jointly serves the users' demands is different from the existing works that divide the multi-request problem into a sequence of single-request problems. We can also apply the proposed strategy to the coded caching problem with multiple requests. By doing so, we can give an optimal scheme for the four cases that were left open in [35] for the coded caching problem with multiple requests, where the setting includes up to four users and each user with memory size $\mathsf{M} = \mathsf{N}/\mathsf{K}$ demands at most two files. The details of how to modify our proposed scheme (so as to account for the lack of symmetry of the multi-request problem) are given in Appendix G.

4) Extension to Coded Distributed Computing: When $\mathsf{N} = \mathsf{K}$ and each user demands a distinct file, the $(\mathsf{N}, \mathsf{K}, \mathsf{r}, \mathsf{M})$ shared-link caching problem with correlated files of combinatorial overlaps is related to the coded distributed computing problem in [29]. The only difference is that in [29] the link is D2D (i.e., workers/users communication among each other without a central master/server), as opposed to the shared-link case (with a central server broadcasting messages) considered here. In [29], the authors proposed an optimal scheme that requires to exchange messages where symbols are from a large finite field size. In contrast, for the shared-link case, the proposed scheme for Theorem 2 is optimal, whose operations are simpler in that they are on the binary field.

G. Numerical Evaluations

In the following, we provide some numerical evaluations to illustrate the proposed converse and achievable bounds, which are also compared with the achievable bound in [21].

In Fig. 1, we consider the $(\mathsf{N}, \mathsf{K}, \mathsf{r}) = (20, 20, 5)$ shared-link caching problem with correlated files of combinatorial overlaps for distinct demands. In Fig. 2, we consider the $(\mathsf{N}, \mathsf{K}, \mathsf{r}) = (20, 30, 2)$ problem, where in Fig. 2a we plot the average load for the demand type $\mathcal{D}_{20}$ and in Fig. 2b we plot the average load over all possible demands. It can be seen from in Fig. 1 and Fig. 2 that the proposed schemed coincide

$$e^s_t := \frac{\sum_{j \in [\min\{s, \mathsf{N}-\mathsf{r}+1, \mathsf{K}-t\}]} \sum_{q=j+1}^{\min\{\mathsf{N}-\mathsf{r}+2, \mathsf{K}-t+1, s\}} \left(\binom{\mathsf{N}-q}{\mathsf{r}-2} - \binom{\mathsf{N}-s}{\mathsf{r}-2} \right) \left(\binom{\mathsf{K}-q}{t-1} - \binom{\mathsf{K}-s}{t-1} \right)}{\binom{\mathsf{N}-1}{\mathsf{r}-1} \binom{\mathsf{K}}{t}}. \tag{13}$$

Authorized licensed use limited to: University of Illinois at Chicago Library. Downloaded on November 08,2023 at 23:24:46 UTC from IEEE Xplore. Restrictions apply.

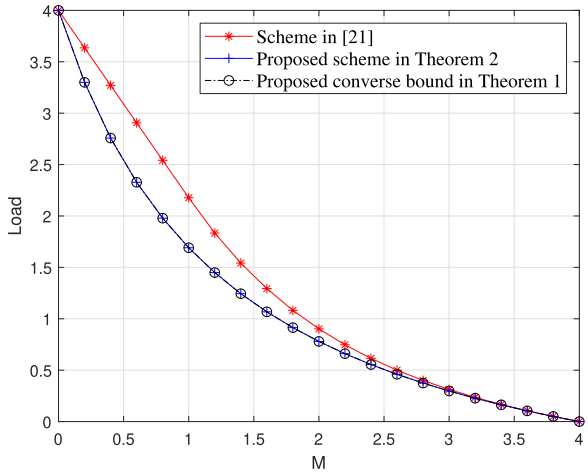
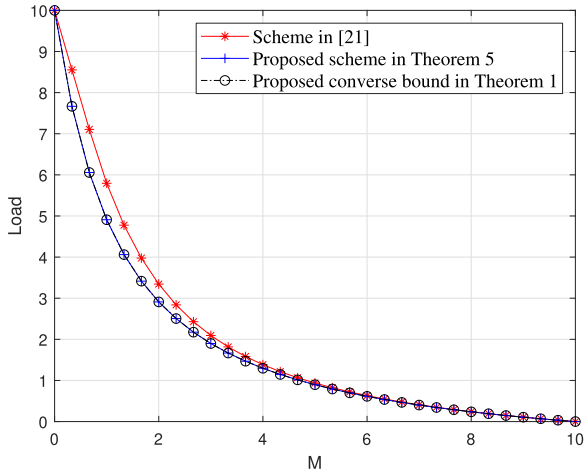
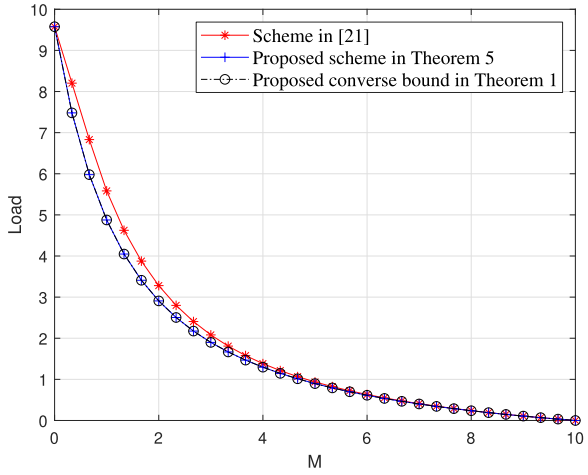


Fig. 1. The memory-load trade-off for the $(\mathsf{N},\mathsf{K},\mathsf{r}) = (20,20,5)$ shared-link caching problem with correlated files of combinatorial overlaps for distinct demands.



(a) Demand type $\mathcal{D}_{20}$.



(b) Average load over all demands.

Fig. 2. The memory-load trade-off for the $(\mathsf{N},\mathsf{K},\mathsf{r}) = (20,30,2)$ shared-link caching problem with correlated files of combinatorial overlaps.

with the proposed converse bound, and outperform the scheme in [21] for all memory sizes.

In Fig. 3, we plot the average load for the demand type $\mathcal{D}_{20}$ (Fig. 3a) and the average load over all possible demands (Fig. 3b) for the $(\mathsf{N},\mathsf{K},\mathsf{r}) = (20,30,3)$ shared-link caching problem with correlated files of combinatorial overlaps. When $t = \mathsf{KMr}/\mathsf{N} \in \{0,1,2,29,30\}$ or $N_{\mathrm{e}}(\mathbf{d}) \leq 4$, only the first sub-phase of the delivery scheme is necessary as stated in Theorem 5. For other values of the parameters t and $N_{\mathrm{e}}(\mathbf{d})$, we use both sub-phases of the delivery scheme as stated in Theorem 3. Fig. 3 shows that our proposed achievable schemes outperform the scheme in [21] for a large range of memory sizes.

IV. Converse Bound

Under the constraint of uncoded cache placement, we can partition each block $W_{\mathcal{S}}$ where $\mathcal{S} \subseteq [\mathsf{N}]$ and $|\mathcal{S}| = \mathsf{r}$, into sub-blocks as

$$W_{\mathcal{S}} = \{W_{\mathcal{S},\mathcal{V}} : \mathcal{V} \subseteq [\mathsf{K}]\},\ \forall \mathcal{S} \subseteq [\mathsf{N}] : |\mathcal{S}| = \mathsf{r}, \tag{21}$$

where $W_{\mathcal{S},\mathcal{V}}$ represents the bits of $W_{\mathcal{S}}$ exclusively cached by users indexed by $\mathcal{V}$.[6]

A. Proof of Theorem 1

The delivery phase with uncoded cache placement is equivalent to a multicast index coding problem [47]. Such a problem can be represented on a directed graph. In this graph, each sub-block demanded but not cached by a user is a node; a directed edge exists from node a to node b if the user demanding the sub-block represented by node b has the sub-block represented by node a in its cache. As in [3], we use the acyclic index coding converse bound from [46] to lower bound the number of transmitted bits needed to satisfy all the nodes/users in this index coding problem as follows.

For a demand vector $\mathbf{d}$ with $N_{\mathrm{e}}(\mathbf{d})$ demanded files, we choose $N_{\mathrm{e}}(\mathbf{d})$ users (i.e., leaders) each of which demands a distinct file. We then draw a graph where each sub-block demanded but not cached by some of these $N_{\mathrm{e}}(\mathbf{d})$ users is a node. We then consider a permutation of these $N_{\mathrm{e}}(\mathbf{d})$ users, denoted by $\mathbf{u} = (u_1, u_2, \ldots, u_{N_{\mathrm{e}}(\mathbf{d})})$. The set of sub-blocks

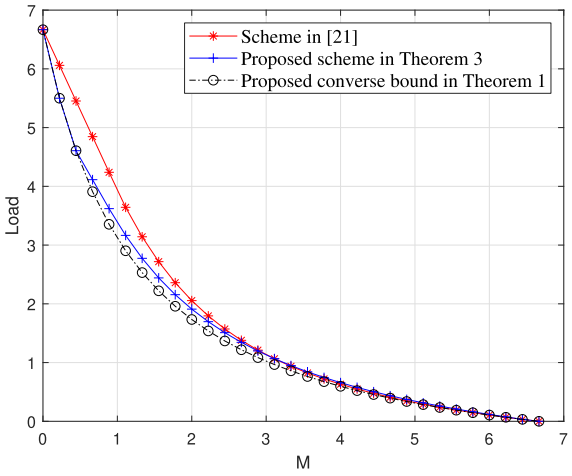
$$\bigcup_{k\in[\min\{N_{\mathrm{e}}(\mathbf{d}),\mathsf{N}-\mathsf{r}+1\}]}\ \bigcup_{\substack{\mathcal{S}\subseteq[\mathsf{N}]\setminus\{d_{u_1},\ldots,d_{u_{k-1}}\}:\\ |\mathcal{S}|=\mathsf{r},d_{u_k}\in\mathcal{S}}}\ \bigcup_{\mathcal{V}\subseteq[\mathsf{K}]\setminus\{u_1,\ldots,u_k\}} W_{\mathcal{S},\mathcal{V}}, \tag{22}$$

does not contain a directed cycle.[7] This can be seen as follows, similar to [3, Lemma 1]. We classify the sub-blocks/nodes in the set (22) into levels. More precisely, we say that sub-block/node $W_{\mathcal{S},\mathcal{V}}$ is in level i if $\mathcal{S} \subseteq [\mathsf{N}] \setminus \{d_{u_1},\ldots,d_{u_{i-1}}\}$, $d_{u_i} \in \mathcal{S}$ and $\mathcal{V} \subseteq [\mathsf{K}] \setminus \{u_1,\ldots,u_i\}$. Each node in level i is a sub-block that is demanded by user u_i and is not cached by any user in $\{u_1,\ldots,u_i\}$, and corresponds to a user in the index coding problem that has the same side information as user u_i in our caching problem (i.e., each node in level i only

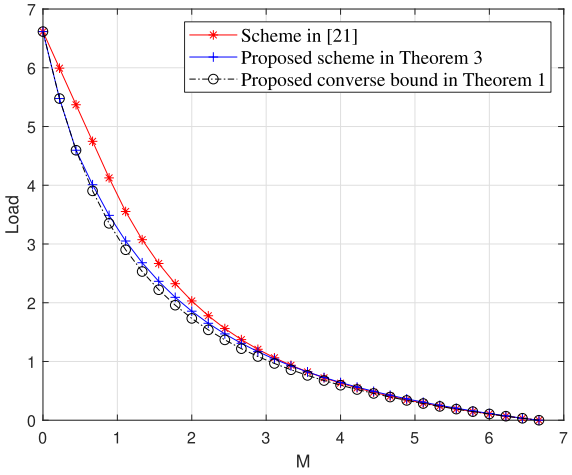
[6]Note that $\mathcal{V}$ can be the empty set and $W_{\mathcal{S},\emptyset}$ represents the bits of $W_{\mathcal{S}}$ not cached by any user.

[7]Note that in (22), k should be no more than $\mathsf{N}-\mathsf{r}+1$. This is because, $d_{u_1},\ldots,d_{u_{k-1}}$ are distinct; thus, if $k > \mathsf{N}-\mathsf{r}+1$, there does not exist such $\mathcal{S} \subseteq [\mathsf{N}] \setminus \{d_{u_1},\ldots,d_{u_{k-1}}\}$ where $|\mathcal{S}| = \mathsf{r}$.

Authorized licensed use limited to: University of Illinois at Chicago Library. Downloaded on November 08,2023 at 23:24:46 UTC from IEEE Xplore. Restrictions apply.



(a) Demand type $\mathcal{D}_{20}$.



(b) Average load over all demands.

Fig. 3. The memory-load trade-off for the $(\mathsf{N},\mathsf{K},\mathsf{r}) = (20, 30, 3)$ shared-link caching problem with correlated files of combinatorial overlaps.

knows the nodes $W_{\mathcal{S},\mathcal{V}}$ where $u_i \in \mathcal{V}$). So each node in level i knows neither the nodes in the same level, nor the nodes in the higher levels. As a result, the proposed set in (22) does not contain a directed cycle.

By the acyclic index coding converse bound, the number of transmitted bits is not less than the total number of bits of the sub-blocks in the set in (22); that is,

$$\mathsf{R}_{\mathrm{u}}^{\star}(\mathsf{M}, N_{\mathrm{e}}(\mathbf{d})) \geq \sum_{k\in[\min\{N_{\mathrm{e}}(\mathbf{d}),\mathsf{N}-\mathsf{r}+1\}]} \sum_{\substack{\mathcal{S}\subseteq[\mathsf{N}]\setminus\{d_{u_1},\ldots,d_{u_{k-1}}\}:\\ |\mathcal{S}|=\mathsf{r},d_{u_k}\in\mathcal{S}}} \sum_{\mathcal{V}\subseteq[\mathsf{K}]\setminus\{u_1,\ldots,u_k\}} \frac{|W_{\mathcal{S},\mathcal{V}}|}{\mathsf{B}}, \tag{23}$$

where $|W_{\mathcal{S},\mathcal{V}}|$ represents the length of $W_{\mathcal{S},\mathcal{V}}$ in bits.

For a fixed $s \in [\min\{\mathsf{K},\mathsf{N}\}]$, consider all the demands of type $\mathcal{D}_s$, all sets of s users with different s distinct demands, and all permutations of those users. By summing together all the resulting inequalities as in (23) we obtain the lower bound on $\mathsf{R}_{\mathrm{u}}^{\star}(\mathsf{M}, s)$ in (24), shown at the bottom of the next page. To obtain (24), the number of possible leader sets including s leaders equals $\binom{\mathsf{K}}{s}$, the number of demand vectors where a set of s users can serve as leader set equals $\frac{\mathsf{N}!\, s^{\mathsf{K}-s}}{(\mathsf{N}-s)!}$, the number of permutations of a leader set equals $s!$. By the symmetry of the problem, in (24) the coefficient of each sub-block stored by exactly t users is identical, for each $t \in [0:\mathsf{K}]$. In (24), there are $\sum_{j\in[\min\{s,\mathsf{N}-\mathsf{r}+1,\mathsf{K}-t\}]} \binom{\mathsf{N}-j}{\mathsf{r}-1}\binom{\mathsf{K}-j}{t}$ sub-blocks known by exactly t users. We also note that in the whole library there are $\binom{\mathsf{N}}{\mathsf{r}}\binom{\mathsf{K}}{t}$ sub-blocks known by exactly t users. Hence, we have

$$\begin{aligned}
\mathsf{R}_{\mathrm{u}}^{\star}(\mathsf{M}, s) &\geq \sum_{t=0}^{\mathsf{K}} \left(\frac{\sum_{j\in[\min\{s,\mathsf{N}-\mathsf{r}+1,\mathsf{K}-t\}]} \binom{\mathsf{N}-j}{\mathsf{r}-1}\binom{\mathsf{K}-j}{t}}{\binom{\mathsf{N}}{\mathsf{r}}\binom{\mathsf{K}}{t}} \right. \\
&\qquad \left. \cdot \frac{\sum_{\mathcal{S}\subseteq[\mathsf{N}]:|\mathcal{S}|=\mathsf{r}} \sum_{\mathcal{V}\subseteq[\mathsf{K}]:|\mathcal{V}|=t} |W_{\mathcal{S},\mathcal{V}}|}{\mathsf{B}} \right) \\
&= \sum_{t=0}^{\mathsf{K}} \left(\frac{\sum_{j\in[\min\{s,\mathsf{N}-\mathsf{r}+1,\mathsf{K}-t\}]} \binom{\mathsf{N}-j}{\mathsf{r}-1}\binom{\mathsf{K}-j}{t}}{\binom{\mathsf{N}-1}{\mathsf{r}-1}\binom{\mathsf{K}}{t}} \right. \\
&\qquad \left. \cdot \frac{\sum_{\mathcal{S}\subseteq[\mathsf{N}]:|\mathcal{S}|=\mathsf{r}} \sum_{\mathcal{V}\subseteq[\mathsf{K}]:|\mathcal{V}|=t} \mathsf{r}|W_{\mathcal{S},\mathcal{V}}|}{\mathsf{NB}} \right) \\
&= \sum_{t=0}^{\mathsf{K}} c_t^s \cdot x_t, \qquad (25a)
\end{aligned}$$

$$x_t := \sum_{\mathcal{S}\subseteq[\mathsf{N}]:|\mathcal{S}|=\mathsf{r}} \sum_{\mathcal{V}\subseteq[\mathsf{K}]:|\mathcal{V}|=t} \frac{\mathsf{r}|W_{\mathcal{S},\mathcal{V}}|}{\mathsf{NB}}, \tag{25b}$$

$$c_t^s := \frac{\sum_{j\in[\min\{s,\mathsf{N}-\mathsf{r}+1,\mathsf{K}-t\}]} \binom{\mathsf{N}-j}{\mathsf{r}-1}\binom{\mathsf{K}-j}{t}}{\binom{\mathsf{N}-1}{\mathsf{r}-1}\binom{\mathsf{K}}{t}},$$

(as already defined in (9)), (25c)

$$x_0 + x_1 + \ldots + x_{\mathsf{K}} = 1, \text{ (file size constraint)}, \tag{25d}$$

$$x_1 + 2x_2 + \ldots + tx_t + \ldots + \mathsf{K}x_{\mathsf{K}} \leq \frac{\mathsf{KMr}}{\mathsf{N}},$$

(memory size contraint), (25e)

where x_t in (25b) represents the fraction of all the bits in the library that are cached exactly by t users.

As in [4], we can lower bound (25a) by using Jensen's inequality and the monotonicity of $\mathrm{Conv}(c_t^s)$ (i.e., the convex lower envelope of c_t^s in terms of t),

$$\mathsf{R}_{\mathrm{u}}^{\star}(\mathsf{M}, s) \geq \mathrm{Conv}(c_t^s). \tag{26}$$

By considering all the demand types, and from (26), we also have

$$\mathsf{R}_{\mathrm{u}}^{\star}(\mathsf{M}) \geq \mathbb{E}_{s\in[\min\{\mathsf{N},\mathsf{K}\}]}\left[\mathsf{R}_{\mathrm{u}}^{\star}(\mathsf{M}, s)\right] \geq \mathbb{E}_{s\in[\min\{\mathsf{N},\mathsf{K}\}]}[\mathrm{Conv}(c_t^s)]. \tag{27}$$

Since c_t^s is convex in t, we can change the order of the expectation and the 'Conv' in (27). Thus we prove the converse bound in Theorem 1.

Notice that we could also use Fourier-Motzkin elimination to eliminate the parameters $\{x_t\}_{t\in[0:\mathsf{K}]}$ in (25a) and derive the bound in (26), as done in [3].

B. Discussion

We conclude this session with some observations on the proposed converse bound, which we shall use as a guideline to design our achievable scheme.

Authorized licensed use limited to: University of Illinois at Chicago Library. Downloaded on November 08,2023 at 23:24:46 UTC from IEEE Xplore. Restrictions apply.

1) The corner points of our converse bound are of the form $\left(\frac{\mathsf{N}t}{\mathsf{Kr}}, c_t^s\right)$, where c_t^s is defined in (9). The converse bound may suggest the following placement. We partition each block $W_{\mathcal{S}}$ into $\binom{\mathsf{K}}{t}$ equal-length sub-blocks of length $\frac{\mathsf{B}}{\binom{\mathsf{N}-1}{\mathsf{r}-1}\binom{\mathsf{K}}{t}}$ and indicate $W_{\mathcal{S}} = \{W_{\mathcal{S},\mathcal{V}} : \mathcal{V} \subseteq [\mathsf{K}], |\mathcal{V}| = t\}$. Each user $k \in [\mathsf{K}]$ stores the sub-block $W_{\mathcal{S},\mathcal{V}}$ if $k \in \mathcal{V}$. Hence, user $k \in [\mathsf{K}]$ caches $\frac{\mathsf{B}\binom{\mathsf{K}-1}{t-1}\binom{\mathsf{N}}{\mathsf{r}}}{\binom{\mathsf{N}-1}{\mathsf{r}-1}\binom{\mathsf{K}}{t}} = \frac{\mathsf{BN}t}{\mathsf{Kr}}$ bits in total.
We will use this interpretation to design the file partitioning and the cache placement of our proposed caching scheme, which is the same as in [21].
2) If the above placement is used, each sub-block is cached by t users. In the proof of Theorem 1, for each demand $\mathbf{d}$, we choose a set of leader users (each demanding a different file) and consider a permutation $\mathbf{u} = (u_1, \ldots, u_{N_{\mathrm{e}}(\mathbf{d})})$ of these $N_{\mathrm{e}}(\mathbf{d})$ leader users. For the permutation $\mathbf{u}$, we find an acyclic set of $\sum_{j\in[\min\{N_{\mathrm{e}}(\mathbf{d}),\mathsf{N}-\mathsf{r}+1,\mathsf{K}-t\}]}\binom{\mathsf{N}-j}{\mathsf{r}-1}\binom{\mathsf{K}-j}{t}$ sub-blocks, and the load is lower bounded by the total length of these sub-blocks. In addition, in this acyclic set of sub-blocks, there are $\binom{\mathsf{N}-j}{\mathsf{r}-1}\binom{\mathsf{K}-j}{t}$ sub-blocks desired by user u_j where $j \in [\min\{N_{\mathrm{e}}(\mathbf{d}), \mathsf{N}-\mathsf{r}+1, \mathsf{K}-t\}]$; these sub-blocks are not cached nor desired by any user u_{j_1} where $j_1 < j$. This may suggest a delivery scheme with $\min\{N_{\mathrm{e}}(\mathbf{d}), \mathsf{N}-\mathsf{r}+1, \mathsf{K}-t\}$ steps, where in Step j we transmit $\binom{\mathsf{N}-j}{\mathsf{r}-1}\binom{\mathsf{K}-j}{t}$ linear combinations such that each linear combination contains one of the $\binom{\mathsf{N}-j}{\mathsf{r}-1}\binom{\mathsf{K}-j}{t}$ sub-blocks desired by user u_j, and thus at the end of this step user u_j is satisfied.
We will use this interpretation to design the first sub-phase of our general delivery scheme in (11), which we shall introduce next in Section V.

V. Achievable Schemes

In this section, we focus on the achievable scheme in (11) and prove the statements of Theorems 2, 3 and 5. Notice that when $\mathsf{r} \in \{1, \mathsf{N}\}$, the considered problem is equivalent to the MAN problem (solved under the constraint of uncoded cache placement in [4]). Hence, the novelty of our scheme is for $\mathsf{r} \in [2 : \mathsf{N}-1]$. The scheme we propose was summarized in (11); Theorems 2 and 5 only use the first sub-phase of the delivery, while Theorem 3 uses both sub-phases.

The rest of this section is organized as follows. In Section V-A we give an example of the first sub-phase of the proposed delivery scheme in (11); the objective is to highlight how the multicast messages sent in sub-phase 1 enable all leaders to decode their desired file. Then in Section V-B we show which user can decode which sub-blocks after receiving the multicast messages in sub-phase 1, regardless of the demand type. In Section V-C we show that every user can decode its desired message by also receiving the multicast messages sent in sub-phase 2. In Section V-D we give an example of the second sub-phase of the proposed delivery scheme in (11). In Section V-E we prove the order optimality results in Theorems 4 for general case. Finally, in Section V-F we prove the exact optimality results in Theorem 5 by observing each non-leader can reconstruct its required multicast messages in sub-phase 2 by performing linear combinations of the received multicast messages in sub-phase 1.

A. An Example of (11) With Only Sub-Phase 1 for the Delivery Scheme

First, we study an example where $\mathsf{N} \geq \mathsf{K}$ and each user demands a distinct file (i.e., $s = \mathsf{K}$). In particular, we consider the $(\mathsf{N}, \mathsf{K}, \mathsf{r}, \mathsf{M}) = (4, 4, 2, 1/2)$ shared-link caching problem with correlated files of combinatorial overlaps. There are $\binom{\mathsf{N}}{\mathsf{r}} = 6$ blocks denoted as $W_{\{1,2\}}$, $W_{\{1,3\}}$, $W_{\{1,4\}}$, $W_{\{2,3\}}$, $W_{\{2,4\}}$, and $W_{\{3,4\}}$. The files are

$$\begin{aligned} F_1 &= \{W_{\{1,2\}}, W_{\{1,3\}}, W_{\{1,4\}}\},\\ F_2 &= \{W_{\{1,2\}}, W_{\{2,3\}}, W_{\{2,4\}}\},\\ F_3 &= \{W_{\{1,3\}}, W_{\{2,3\}}, W_{\{3,4\}}\},\\ F_4 &= \{W_{\{1,4\}}, W_{\{2,4\}}, W_{\{3,4\}}\}. \end{aligned}$$

1) Block Subdivision: Here $t = \frac{\mathsf{KMr}}{\mathsf{N}} = 1$. We partition each block into $\binom{\mathsf{K}}{t} = 4$ equal-length sub-blocks and denote $W_{\mathcal{S}} = \{W_{\mathcal{S},\mathcal{V}} : \mathcal{V} \subseteq [\mathsf{K}], |\mathcal{V}| = t = 1\} = \{W_{\mathcal{S},\{k\}} : k \in [\mathsf{K}]\}$. Hence, each sub-block contains $\mathsf{B}/\left(\binom{\mathsf{N}-1}{\mathsf{r}-1}\binom{\mathsf{K}}{t}\right) = \mathsf{B}/12$ bits.

2) Placement Phase: The cache placement is inspired by the converse bound (see discussion in Section IV-B). User $k \in [\mathsf{K}]$ caches $W_{\mathcal{S},\mathcal{V}}$ if $k \in \mathcal{V}$; that is, $Z_k = \{W_{\mathcal{S},\{k\}}, \forall \mathcal{S} \subseteq [\mathsf{N}] : |\mathcal{S}| = \mathsf{r} = 2\}$.

3) Delivery Phase: Assume $\mathbf{d} = (1, 2, 3, 4)$, which has $N_{\mathrm{e}}(\mathbf{d}) = 4$ distinct demanded files. Pick one user demanding a distinct file, and refer to it as the leader among those users demanding the same file. Since each user has a distinct request in this example, each user is a leader, and the leader set is $[1 : 4]$. Consider a permutation u of the leaders, say $\mathbf{u} = (u_1, u_2, u_3, u_4) = (1, 2, 3, 4)$.

Our proposed first sub-phase of the general delivery scheme contains $\min\{N_{\mathrm{e}}(\mathbf{d}), \mathsf{N}-\mathsf{r}+1, \mathsf{K}-t\} = 3$ steps; after the j^{th} step, the j^{th} element/leader in the permutation can decode its desired file; after finishing all steps, the remaining leaders can also decode their desired file. We next describe, one by one, the

$$\mathsf{R}_{\mathrm{u}}^{\star}(\mathsf{M}, s) \geq \frac{1}{\binom{\mathsf{K}}{s}} \sum_{\mathcal{L}\subseteq[\mathsf{K}]:|\mathcal{L}|=s} \frac{(\mathsf{N}-s)!}{\mathsf{N}!\, s^{\mathsf{K}-s}} \sum_{\mathbf{d}\in\mathcal{D}_s:\mathcal{L} \text{ are leaders}} \frac{1}{s!} \sum_{\mathrm{u}\in\{\text{permutations of } \mathcal{L}\}} \sum_{k\in[\min(s,\mathsf{N}-\mathsf{r}+1)]} \sum_{\substack{\mathcal{S}\subseteq[\mathsf{N}]\setminus\{d_{u_1},\ldots,d_{u_{k-1}}\}:\\ |\mathcal{S}|=\mathsf{r}, d_{u_k}\in\mathcal{S}}} \sum_{t=0}^{\mathsf{K}-k} \sum_{\mathcal{V}\subseteq[\mathsf{K}]\setminus\{u_1,\ldots,u_k\}:|\mathcal{V}|=t} \frac{|W_{\mathcal{S},\mathcal{V}}|}{\mathsf{B}}; \tag{24}$$

Authorized licensed use limited to: University of Illinois at Chicago Library. Downloaded on November 08,2023 at 23:24:46 UTC from IEEE Xplore. Restrictions apply.

three steps for this example. Recalling (11f), the transmitted multicast messages are of the type

$$C_{\mathcal{J},\mathcal{B}} := \underset{k\in\mathcal{J}}{\oplus} \underset{\substack{\mathcal{S}\subseteq\mathcal{N}_{\mathbf{d}}(\mathcal{J})\cup\mathcal{B}:\\ |\mathcal{S}|=\mathsf{r},\mathcal{B}\subseteq\mathcal{S},d_k\in\mathcal{S}}}{\oplus} W_{\mathcal{S},\mathcal{J}\setminus\{k\}}, \tag{28}$$

for all $\mathcal{J}\subseteq[\mathsf{K}]\setminus\{u_1,\ldots,u_{j-1}\}$ where $|\mathcal{J}|=t+1$, and $u_j\in\mathcal{J}$, and all $\mathcal{B}\subseteq[\mathsf{N}]\setminus\{d_{u_1},\ldots,d_{u_j}\}$ where $|\mathcal{B}|=\mathsf{r}-1$. Note that $\mathcal{N}_{\mathbf{d}}(\mathcal{J})$ is the set of demanded files by the users in $\mathcal{J}$. In (28), the role of $\mathcal{J}$ is to select whose demanded blocks are included in the sum $C_{\mathcal{J},\mathcal{B}}$ and the role of $\mathcal{B}$ is to select which demanded blocks by the users in $\mathcal{J}$ are included into the sum. In plain words, the multicast message $C_{\mathcal{J},\mathcal{B}}$ in (28) is the XOR of sub-blocks $W_{\mathcal{S},\mathcal{J}\setminus\{k\}}$, where $k\in\mathcal{J}$, $\mathcal{S}\subseteq(\mathcal{N}_{\mathbf{d}}(\mathcal{J})\cup\mathcal{B})$, $d_k\in\mathcal{S}$ and $\mathcal{B}\subseteq\mathcal{S}$. Note that, when $\mathsf{r}=1$ (in which case our model reduces to the MAN system in [2]), $C_{\mathcal{J},\emptyset}$ in (28) is equivalent to the MAN multicast message

$$C_{\mathcal{J},\emptyset}=\underset{k\in\mathcal{J}}{\oplus} W_{\{d_k\},\mathcal{J}\setminus\{k\}}. \tag{29}$$

Delivery sub-phase 1.*Step* 1. In this step we aim to satisfy leader user $u_1=1$, who misses three sub-blocks of the three blocks in F_1 (recall that $d_{u_1}=1$); that is, user 1 must recover nine sub-blocks. Each time we consider one set of users $\mathcal{J}\subseteq[\mathsf{K}]$ where $|\mathcal{J}|=t+1=2$ and $u_1\in\mathcal{J}$ (recall that $u_1=1$), and one set of files $\mathcal{B}\subseteq[\mathsf{N}]\setminus\{d_{u_1}\}$ where $|\mathcal{B}|=\mathsf{r}-1=1$, and transmit $C_{\mathcal{J},\mathcal{B}}$. For example, for $\mathcal{J}=\{1,2\}$ and $\mathcal{B}=\{2\}$, we transmit

$$C_{\{1,2\},\{2\}}=W_{\{1,2\},\{2\}}\oplus W_{\{1,2\},\{1\}}. \tag{30a}$$

In $C_{\{1,2\},\{2\}}$, user 1 knows $W_{\{1,2\},\{1\}}$ and can thus decode $W_{\{1,2\},\{2\}}$. Similarly, user 2 knows $W_{\{1,2\},\{2\}}$ and can thus decode $W_{\{1,2\},\{1\}}$. Similarly, we transmit

$$C_{\{1,2\},\{3\}}=W_{\{1,3\},\{2\}}\oplus W_{\{2,3\},\{1\}}; \tag{30b}$$
$$C_{\{1,2\},\{4\}}=W_{\{1,4\},\{2\}}\oplus W_{\{2,4\},\{1\}}; \tag{30c}$$
$$C_{\{1,3\},\{2\}}=W_{\{1,2\},\{3\}}\oplus W_{\{2,3\},\{1\}}; \tag{30d}$$
$$C_{\{1,3\},\{3\}}=W_{\{1,3\},\{3\}}\oplus W_{\{1,3\},\{1\}}; \tag{30e}$$
$$C_{\{1,3\},\{4\}}=W_{\{1,4\},\{3\}}\oplus W_{\{3,4\},\{1\}}; \tag{30f}$$
$$C_{\{1,4\},\{2\}}=W_{\{1,2\},\{4\}}\oplus W_{\{2,4\},\{1\}}; \tag{30g}$$
$$C_{\{1,4\},\{3\}}=W_{\{1,3\},\{4\}}\oplus W_{\{3,4\},\{1\}}; \tag{30h}$$
$$C_{\{1,4\},\{4\}}=W_{\{1,4\},\{4\}}\oplus W_{\{1,4\},\{1\}}. \tag{30i}$$

From (30) and its cached content, user $u_1=1$ can recover $W_{\{1,2\}}$, $W_{\{1,3\}}$, and $W_{\{1,4\}}$. Thus, user 1 is satisfied after this step (i.e., it has recovered the missing nine sub-blocks from the nine received multicast messages in the first step).

Let us then focus on user $u_2=2$. User 2 can directly recover $W_{\{1,2\},\{1\}}$ from (30a), $W_{\{2,3\},\{1\}}$ from (30b), $W_{\{2,4\},\{1\}}$ from (30c). Since user 2 has recovered $W_{\{2,3\},\{1\}}$, it then can recover $W_{\{1,2\},\{3\}}$ from (30d). Since user 2 has recovered $W_{\{2,4\},\{1\}}$, it then can recover $W_{\{1,2\},\{4\}}$ from (30g). In conclusion, after Step 1, user 2 can recover $W_{\{1,2\}}$ and also recover $W_{\{2,3\},\{1\}}$ and $W_{\{2,4\},\{1\}}$. User $u_2=2$ after Step 1 still misses four sub-blocks, namely $\{W_{\{2,3\},\{k\}},W_{\{2,4\},\{k\}}:k\in[3,4]\}$.

Similar to user $u_2=2$, each user $k\in\{3,4\}$ can recover $W_{\mathcal{S}}$ where $\{d_{u_1},d_k\}\subseteq\mathcal{S}$, and can also recover $W_{\mathcal{S}_1,\mathcal{V}_1}$ where $d_k\in\mathcal{S}_1$ and $u_1\in\mathcal{V}_1$ after Step 1. Each of these users still misses four sub-blocks after Step 1.

Delivery sub-phase 1.*Step* 2. In this step we aim to satisfy leader user $u_2=2$. Each time we consider one set of users $\mathcal{J}\subseteq([\mathsf{K}]\setminus\{u_1\})$ where $|\mathcal{J}|=t+1$ and $u_2\in\mathcal{J}$, and one set of files $\mathcal{B}\subseteq([\mathsf{N}]\setminus\{d_{u_1},d_{u_2}\})$ where $|\mathcal{B}|=\mathsf{r}-1=1$ (recall that $u_1=d_{u_1}=1,u_2=d_{u_2}=2$), and transmit $C_{\mathcal{J},\mathcal{B}}$. For example, for $\mathcal{J}=\{2,3\}$ and $\mathcal{B}=\{3\}$, we transmit

$$C_{\{2,3\},\{3\}}=W_{\{2,3\},\{3\}}\oplus W_{\{2,3\},\{2\}}. \tag{31a}$$

From (31a), user 2 can recover $W_{\{2,3\},\{3\}}$ and user 3 can recover $W_{\{2,3\},\{2\}}$. Similarly, we transmit

$$C_{\{2,3\},\{4\}}=W_{\{2,4\},\{3\}}\oplus W_{\{3,4\},\{2\}}; \tag{31b}$$
$$C_{\{2,4\},\{3\}}=W_{\{2,3\},\{4\}}\oplus W_{\{3,4\},\{2\}}; \tag{31c}$$
$$C_{\{2,4\},\{4\}}=W_{\{2,4\},\{4\}}\oplus W_{\{2,4\},\{2\}}. \tag{31d}$$

From (31) user $u_2=2$ can recover the desired sub-blocks that were not recovered from Step 1. User $u_2=2$ is satisfied after this step (i.e., it has recovered the missing four sub-blocks from the four received multicast messages in the second step).

Let us then focus on user $u_3=3$. User 3 can directly recover $W_{\{2,3\},\{2\}}$ from (31a) and $W_{\{3,4\},\{2\}}$ from (31b). Since user 3 has recovered $W_{\{3,4\},\{2\}}$, it then can recover $W_{\{2,3\},\{4\}}$ from (31c). User $u_3=3$ after this step still misses $W_{\{3,4\},\{4\}}$.

Similar to user $u_3=3$, at the end of Step 2, user $k=4$ can recover $W_{\mathcal{S}}$ where $d_k\in\mathcal{S}$, $\{d_{u_1},d_{u_2}\}\cap\mathcal{S}\neq\emptyset$, and also recover $W_{\mathcal{S}_1,\mathcal{V}_1}$ where $d_k\in\mathcal{S}_1$ and $\{u_1,u_2\}\cap\mathcal{V}_1\neq\emptyset$. User 4 still misses one sub-block ($W_{\{3,4\},\{3\}}$) after Step 2.

Delivery sub-phase 1.*Step* 3. In this step we aim to satisfy leader user $u_3=3$. Each time we consider one set of users $\mathcal{J}\subseteq([\mathsf{K}]\setminus\{u_1,u_2\})$ where $|\mathcal{J}|=t+1$ and $u_3\in\mathcal{J}$, and one set of files $\mathcal{B}\subseteq([\mathsf{N}]\setminus\{d_{u_1},d_{u_2},d_{u_3}\})$ where $|\mathcal{B}|=\mathsf{r}-1=1$ (recall that $u_1=d_{u_1}=1,u_2=d_{u_2}=2,u_3=d_{u_3}=3$), and transmit $C_{\mathcal{J},\mathcal{B}}$. Hence, at this point there is one possibility, $\mathcal{J}=\{3,4\}$ and $\mathcal{B}=\{4\}$, for which we transmit

$$C_{\{3,4\},\{4\}}=W_{\{3,4\},\{4\}}\oplus W_{\{3,4\},\{3\}}. \tag{32}$$

From (32), user 3 can recover $W_{\{3,4\},\{4\}}$, and user 4 can recover $W_{\{3,4\},\{3\}}$. Hence, at the end of Step 3, users 3 and 4 are satisfied.

4) Performance: Based on the above placement and delivery phases, all users are able to decode their desired blocks. We sent $\sum_{j\in[3]}\binom{\mathsf{N}-j}{\mathsf{r}-1}\binom{\mathsf{K}-j}{t}=14$ linear combinations, each of length $\mathsf{B}/12$ bits. So the load is $7/6$, which coincides with the converse bound in Theorem 1 for $s=4$.

5) Comparison With the State-of-the-Art 'Round-Division' Schemes: Let us then consider the round-division methods in [21], [34], [35], [36], [37], [38], [39], and [40]. If there exists some sub-block appearing in different rounds, the round-division strategy that treats each round as an independent single-request MAN caching problem may miss some multicast opportunities. Here we show that the round-division strategy is sub-optimal even if we can divide the users' demands into multiple rounds such that there does not exist any sub-block appearing in different rounds. More precisely,

Authorized licensed use limited to: University of Illinois at Chicago Library. Downloaded on November 08,2023 at 23:24:46 UTC from IEEE Xplore. Restrictions apply.

since each user demands 3 blocks, we can divide the delivery into the following three rounds:

- Round 1: In the first round, users 1 and 2 demand $W_{\{1,2\}}$, and users 3 and 4 demand $W_{\{3,4\}}$. This is equivalent to the MAN caching problem with 4 users and 2 files. By using the optimal caching scheme under the constraint of uncoded cache placement in [4], we need to transmit $\binom{\mathsf{K}}{t} - \binom{\mathsf{K}-\mathsf{N}}{t} = \binom{4}{2} - \binom{2}{2} = 5$ linear combinations, each of which contains $\mathsf{B}/12$ bits, in order to satisfy these requests.
- Round 2: In the second round, users 1 and 3 demand $W_{\{1,3\}}$, and users 2 and 4 demand $W_{\{2,4\}}$. By using the caching scheme in [4], we need to transmit 5 linear combinations to satisfy these requests.
- Round 3: In the third round, users 1 and 4 demand $W_{\{1,4\}}$, and users 2 and 3 demand $W_{\{2,3\}}$. By using the caching scheme in [4], we need to transmit 5 linear combinations to satisfy these requests.

Hence, by this round-division strategy, the load is $15/12 > 7/6$, which is strictly sub-optimal. In conclusion, in order to achieve optimality in this example, we need to jointly serve the users' demands (as proposed in this paper) in order to fully leverage all multicast opportunities.

B. Proof of Theorem 2

Here we shall prove that after the first sub-phase of the delivery scheme in (11) every leader user is able to decode its desired file (as in the example in Section V-A), and that the load of the first sub-phase (i.e, c_t^s) matches the load of the converse bound in (9) (i.e., $\mathsf{R}_{\mathrm{u}}^{\star}(\mathsf{M},\mathsf{s})$). Thus, for the case where every user is a leader (i.e., every user demands a distinct file), we have proved the exact optimality under the constraint of uncoded cache placement of the proposed achievable scheme as claimed in Theorem 2.

1) Decodability After Delivery Sub-Phase 1: We need to establish which user can decode which sub-blocks at each step of delivery sub-phase 1. The following Lemma 1, which is proved by induction in Appendix A, describes the decoding procedure for delivery sub-phase 1 for general demands (i.e., not only for distinct demands).

Lemma 1 (Decoding After Sub-Phase 1): In the first sub-phase of the proposed delivery scheme in (11) with leader set $\mathcal{L}(\mathbf{d}) = \{u_1,\ldots,u_{N_{\mathrm{e}}(\mathbf{d})}\}$, in Step $j \in [\min\{N_{\mathrm{e}}(\mathbf{d}), \mathsf{N}-\mathsf{r}+1, \mathsf{K}-t\}]$, for each set of users $\mathcal{J}$ where $\mathcal{J} \subseteq [\mathsf{K}] \setminus \{u_1,\ldots,u_{j-1}\}$ such that $|\mathcal{J}| = t+1$ and $u_j \in \mathcal{J}$, and each set of files $\mathcal{B} \subseteq [\mathsf{N}] \setminus \{d_{u_1},\ldots,d_{u_j}\}$ where $|\mathcal{B}| = \mathsf{r}-1$, we transmit $C_{\mathcal{J},\mathcal{B}}$ as defined in (28).

Let $u_{g(i)}$ represent the leader user demanding file F_i, for each $i \in \mathcal{N}_{\mathbf{d}}([\mathsf{K}])$. At the end of the delivery sub-phase 1, we have:

1) For any $C_{\mathcal{J},\mathcal{B}}$ transmitted in sub-phase 1, each user in $\mathcal{J}$ can recover all the sub-blocks in $C_{\mathcal{J},\mathcal{B}}$.
2) At the end of Step $j \in [\min\{g(d_k), \mathsf{N}-\mathsf{r}+1, \mathsf{K}-t\}]$, user $k \in [\mathsf{K}]$ can recover $W_{\mathcal{S},\mathcal{V}}$ if $d_k \in \mathcal{S}$ and $\{u_1,\ldots,u_j\} \cap \mathcal{V} \neq \emptyset$.
3) At the end of Step $j \in [\min\{g(d_k)-1, \mathsf{N}-\mathsf{r}+1, \mathsf{K}-t\}]$, user $k \in [\mathsf{K}]$ can recover $W_{\mathcal{S}}$ if $d_k \in \mathcal{S}$ and $\{d_{u_1},\ldots,d_{u_j}\} \cap \mathcal{S} \neq \emptyset$.

□

2) Decodability for Leader Users After Sub-Phase 1: We use Lemma 1 to show that every leader user is able to recover its demanded file after delivery sub-phase 1. Indeed, under any system parameters, for leader user u_p where $p \in [N_{\mathrm{e}}(\mathbf{d})]$, we have:

- Case $p \leq \min\{N_{\mathrm{e}}(\mathbf{d}), \mathsf{N}-\mathsf{r}+1, \mathsf{K}-t\}$.
  By Lemma 1.Item 3, user u_p can recover $W_{\mathcal{S}}$, where $d_{u_p} \in \mathcal{S}$ and $\{d_{u_1},\ldots,d_{u_{p-1}}\} \cap \mathcal{S} \neq \emptyset$, at the end of Step $p-1$.
  In addition, by Lemma 1.Item 2, user u_p can also recover $W_{\mathcal{S}_1,\mathcal{V}_1}$, where $d_{u_p} \in \mathcal{S}_1$ and $\{u_1,\ldots,u_{p-1}\} \cap \mathcal{V}_1 \neq \emptyset$, at the end of Step $p-1$.
  Hence, user u_p still needs to recover $W_{\mathcal{S}_2,\mathcal{V}_2}$, where $d_{u_p} \in \mathcal{S}_2$, $\{d_{u_1},\ldots,d_{u_{p-1}}\} \cap \mathcal{S}_2 = \emptyset$ and $\{u_1,\ldots,u_p\} \cap \mathcal{V}_2 = \emptyset$. Such a $W_{\mathcal{S}_2,\mathcal{V}_2}$ appears in $C_{\mathcal{V}_2 \cup \{u_p\},\mathcal{S}_2 \setminus \{d_{u_p}\}}$, which is sent in Step p. Hence, by Lemma 1.Item 1, user u_p can recover $W_{\mathcal{S}_2,\mathcal{V}_2}$ at the end of Step p.
- Case $N_{\mathrm{e}}(\mathbf{d}) > \min\{\mathsf{N}-\mathsf{r}+1, \mathsf{K}-t\}$ and $\min\{\mathsf{N}-\mathsf{r}+1, \mathsf{K}-t\} < p \leq N_{\mathrm{e}}(\mathbf{d})$.
  We distinguish two sub-cases:
  - $\mathsf{N}-\mathsf{r}+1 \leq \mathsf{K}-t$. For each desired block of user u_p (assumed to be $W_{\mathcal{S}}$), we have $|\mathcal{S}| = \mathsf{r}$ and thus $\mathcal{S} \cap \{d_{u_1},\ldots,d_{u_{\mathsf{N}-r+1}}\} \neq \emptyset$. Hence, user u_p can recover $W_{\mathcal{S}}$ at the end of the last step (Step $\mathsf{N}-\mathsf{r}+1$) by Lemma 1.Item 3.
  - $\mathsf{N}-\mathsf{r}+1 > \mathsf{K}-t$. For each desired sub-block of user u_p (assumed to be $W_{\mathcal{S}_1,\mathcal{V}_1}$ where $u_p \notin \mathcal{V}_1$), we have $|\mathcal{V}_1| = t$ and thus $\mathcal{V}_1 \cap \{u_1,\ldots,u_{\mathsf{K}-t}\} \neq \emptyset$. Hence, user u_p can recover $W_{\mathcal{S}_1,\mathcal{V}_1}$ at the end of the last step (Step $\mathsf{K}-t$) by Lemma 1.Item 2.

This proves that each leader can recover its demanded file after sub-phase 1.

3) Load of Sub-Phase 1: This proposed sub-phase 1 of the delivery scheme transmits $\binom{\mathsf{N}-j}{\mathsf{r}-1}\binom{\mathsf{K}-j}{t}$ multicast messages in Step $j \in [\min\{N_{\mathrm{e}}(\mathbf{d}), \mathsf{N}-\mathsf{r}+1, \mathsf{K}-t\}]$, which follows the intuition from the proof of our converse bound (see discussion in Section IV-B). Thus, by summing over all steps in sub-phase 1, we get that the load of this delivery sub-phase matches the load of the converse bound in (9).

4) Optimality for the Case of Distinct Demands: From the above reasoning, when all users are leaders (i.e., for the case $\mathsf{N} \geq \mathsf{K}$ and demand type $\mathcal{D}_{\mathsf{K}}$), the claim of Theorem 2 is proved.

C. Proof of Theorem 3

In the following, we focus on general demands and will prove that after the two sub-phases of the delivery scheme in (11), every user is able to decode its desired file. This requires showing that after the second sub-phase the demands of all non-leader users are satisfied. Sub-phase 2 of the delivery scheme in (11) is a form of interference alignment.

The block split and the cache placement phase are as described in (11). The delivery phase contains two sub-phases,

Authorized licensed use limited to: University of Illinois at Chicago Library. Downloaded on November 08,2023 at 23:24:46 UTC from IEEE Xplore. Restrictions apply.

where the first sub-phase is the same as in Section V-B, and the second sub-phase is such that non-leader can align and then cancel the non-demanded sub-blocks, in order to eventually decode their demanded file. We specify next what each user can decode at the end of each step.

1) First Delivery Sub-Phase: In Step $j \in [\min\{N_{\mathrm{e}}(\mathbf{d}), \mathsf{N}-\mathsf{r}+1, \mathsf{K}-t\}]$ of the first sub-phase, for each set of users $\mathcal{J} \subseteq [\mathsf{K}] \setminus \{u_1, \ldots, u_{j-1}\}$ where $|\mathcal{J}| = t+1$ and $u_j \in \mathcal{J}$, and each set of files $\mathcal{B} \subseteq [\mathsf{N}] \setminus \{d_{u_1}, \ldots, d_{u_j}\}$ where $|\mathcal{B}| = \mathsf{r}-1$, we transmit $C_{\mathcal{J},\mathcal{B}}$ as defined in (28). As shown in Section V-B, at the end of this sub-phase, each leader can recover its desired file.

By Lemma 1 (recall that $u_{g(i)}$ represent the leader user who demands file F_i), each non-leader user $k \in [\mathsf{K}] \setminus \mathcal{L}(\mathbf{d})$ can decode $W_{\mathcal{S}}$, where $d_k \in \mathcal{S}$ and $\{d_{u_1}, \ldots, d_{u_{g(d_k)-1}}\} \cap \mathcal{S} \neq \emptyset$, and can decode $W_{\mathcal{S}_1,\mathcal{V}_1}$ where $d_k \in \mathcal{S}_1$ and $\{u_1, \ldots, u_{g(d_k)}\} \cap \mathcal{V}_1 \neq \emptyset$. In addition, for each $j \in [g(d_k)+1 : N_{\mathrm{e}}(\mathbf{d})]$, user k can recover $W_{\{d_k\}\cup\mathcal{S}_2,\{u_j\}\cup\mathcal{V}_2}$ by directly reading off from $C_{\{u_j,k\}\cup\mathcal{V}_2,\mathcal{S}_2}$ (transmitted in Step j of the first sub-phase), where $\mathcal{S}_2 \cap \{d_{u_1}, \ldots, d_{u_j}\} = \emptyset$ and $\mathcal{V}_2 \cap \{u_1, \ldots, u_j\} = \emptyset$.[8]

The non-leader users are thus not yet satisfied, and thus we proceed to send further multicast messages in sub-phase 2.

2) Second Delivery Sub-Phase: The second sub-phase also contains $\min\{N_{\mathrm{e}}(\mathbf{d}), \mathsf{N}-\mathsf{r}+1, \mathsf{K}-t\}$ steps. In Step j, each time we focus on one integer $q \in [j+1 : \min\{\mathsf{N}-\mathsf{r}+2, \mathsf{K}-t+1, N_{\mathrm{e}}(\mathbf{d})\}]$. For each $\mathcal{J} \subseteq ([\mathsf{K}]\setminus\{u_1, \ldots, u_{q-1}\}\cup\{u_j\})$ where $|\mathcal{J}| = t+1$, $\{u_j, u_q\} \subseteq \mathcal{J}$, and $\mathcal{J} \cap \{u_{q+1}, \ldots, u_{N_{\mathrm{e}}(\mathbf{d})}\} \neq \emptyset$, and for each $\mathcal{B} \subseteq [\mathsf{N}] \setminus \{d_{u_1}, \ldots, d_{u_q}\}$ where $|\mathcal{B}| = \mathsf{r}-2$ and $\mathcal{B} \cap \mathcal{N}_{\mathbf{d}}([\mathsf{K}]) \neq \emptyset$, we transmit $C_{\mathcal{J},\mathcal{B}}$ as defined in (28).

In Step j of the second sub-phase, the transmitted multicast message $C_{\mathcal{J},\mathcal{B}}$ by construction satisfies $\mathcal{J} \cap \{u_{q+1}, \ldots, u_{N_{\mathrm{e}}(\mathbf{d})}\} \neq \emptyset$. However, non-leader user k may also need some multicast message(s) $C_{\mathcal{J},\mathcal{B}}$ where $\mathcal{J} \cap \{u_{q+1}, \ldots, u_{N_{\mathrm{e}}(\mathbf{d})}\} = \emptyset$. It is proved in Appendix B that each user k who demands $F_{d_{u_j}}$ can reconstruct $C_{\mathcal{J},\mathcal{B}}$, where $\mathcal{J} \cap \{u_{q+1}, \ldots, u_{N_{\mathrm{e}}(\mathbf{d})}\} = \emptyset$ by using previously received multicast messages, as formalized in the next lemma.

Lemma 2: For each $j \in [\min\{N_{\mathrm{e}}(\mathbf{d}), \mathsf{N}-\mathsf{r}+1, \mathsf{K}-t\}]$, each $q \in [j+1 : \min\{\mathsf{N}-\mathsf{r}+2, \mathsf{K}-t+1, N_{\mathrm{e}}(\mathbf{d})\}]$, each $\mathcal{J} \subseteq [\mathsf{K}] \setminus \mathcal{L}(\mathbf{d}) \cup \{u_j, u_q\}$ where $|\mathcal{J}| = t+1$ and $\{u_j, u_q\} \subseteq \mathcal{J}$, and each $\mathcal{B} \subseteq [\mathsf{N}] \setminus \{d_{u_1}, \ldots, d_{u_q}\}$ where $|\mathcal{B}| = \mathsf{r}-2$ and $\mathcal{B} \cap \mathcal{N}_{\mathbf{d}}([\mathsf{K}]) \neq \emptyset$, user k with $d_k = d_{u_j}$ can reconstruct $C_{\mathcal{J},\mathcal{B}}$ at the end of Step j of sub-phase 2. □

The following Lemma 3, whose proof is in Appendix C, specifies some properties of the linear combinations $C_{\mathcal{J},\mathcal{B}}$ defined in (28).

Lemma 3: [Properties of Function $C_{\mathcal{J},\mathcal{B}}$ Defined in (28)] For each $\mathcal{J} \subseteq [\mathsf{K}]$ where $|\mathcal{J}| = t+1$, and each $\mathcal{B} \subseteq [\mathsf{N}]$ where $|\mathcal{B}| = \mathsf{r}-1$, we have

$$C_{\mathcal{J},\mathcal{B}} = \underset{k\in\mathcal{J}}{\oplus} C_{(\mathcal{J}\setminus\{k\})\cup\{u_{g(i)}\},(\mathcal{B}\setminus\{i\})\cup\{d_k\}}, \tag{33}$$

for any $i \in \mathcal{B}$ where $u_{g(i)} \notin \mathcal{J}$.

[8]This is because in $C_{\{u_j,k\}\cup\mathcal{V}_2,\mathcal{S}_2}$, user k caches all except $W_{\{d_k\}\cup\mathcal{S}_2,\{u_j\}\cup\mathcal{V}_2}$, such that user k can recover $W_{\{d_k\}\cup\mathcal{S}_2,\{u_j\}\cup\mathcal{V}_2}$.

In addition, for each $\mathcal{J} \subseteq [\mathsf{K}]$ where $|\mathcal{J}| = t+1$, and each $\mathcal{B} \subseteq [\mathsf{N}]$ where $|\mathcal{B}| = \mathsf{r}-1$ and $\mathcal{N}_{\mathbf{d}}(\mathcal{J}) \cap \mathcal{B} \neq \emptyset$, we have

$$C_{\mathcal{J},\mathcal{B}} = \underset{i\in\mathcal{N}_{\mathbf{d}}(\mathcal{J})\setminus\mathcal{B}}{\oplus} C_{\mathcal{J},(\mathcal{B}\setminus\{i_1\})\cup\{i\}}, \tag{34}$$

for any $i_1 \in \mathcal{N}_{\mathbf{d}}(\mathcal{J}) \cap \mathcal{B}$. □

By using the transmissions in the two sub-phases and the properties in Lemma 3, we prove the following Lemma 4 (whose proof is in Appendix D), which is the key result for our interference alignment based delivery scheme.

Lemma 4 (Interference Alignment Lemma): For each $j \in [\min\{N_{\mathrm{e}}(\mathbf{d}), \mathsf{N}-\mathsf{r}+1, \mathsf{K}-t\}]$ and each $i \in \{d_{u_1}, \ldots, d_{u_j}\}$, each user can reconstruct $C_{\mathcal{J}\cup\{u_j\},\mathcal{B}\cup\{i\}}$ where $\mathcal{J} \subseteq [\mathsf{K}] \setminus \{u_1, \ldots, u_j\}$, $|\mathcal{J}| = t$, $\mathcal{B} \subseteq [\mathsf{N}] \setminus \{d_{u_1}, \ldots, d_{u_j}\}$, $|\mathcal{B}| = \mathsf{r}-2$, and $\mathcal{N}_{\mathbf{d}}(\mathcal{J}\cap\mathcal{L}(\mathbf{d}))\setminus\mathcal{B} \neq \emptyset$, at the end of Step j of sub-phase 2. □

Lemma 4 can be understood as follows. By Lemma 1, the remaining sub-blocks to be decoded by each non-leader $k \in [\mathsf{K}]\setminus\mathcal{L}(\mathbf{d})$ are $W_{\mathcal{S},\mathcal{V}}$ where $d_k \in \mathcal{S}$, $\{d_{u_1}, \ldots, d_{u_{g(d_k)-1}}\} \cap \mathcal{S} = \emptyset$ and $\{k, u_1, \ldots, u_{g(d_k)}\} \cap \mathcal{V} = \emptyset$. In Step $g(d_k)$ of the first sub-phase, the transmitted message $C_{\mathcal{J},\mathcal{B}}$ should satisfy $d_{g(d_k)} \notin \mathcal{B}$. By Lemma 4, we show that user k can also reconstruct some $C_{\mathcal{J}',\mathcal{B}'}$ where $d_{g(d_k)} \in \mathcal{B}'$. Since $d_{u_{g(d_k)}} \in \mathcal{B}'$, each sub-block in $C_{\mathcal{J}',\mathcal{B}'}$ is desired or cached by user k who demands F_{d_k}. In other words, in order to reconstruct $C_{\mathcal{J}',\mathcal{B}'}$, we align and then cancel the interferences to user k. By induction, all sub-blocks except one in $C_{\mathcal{J}',\mathcal{B}'}$ have been already recovered or cached by user k such that it can recover that sub-block. The detail of the decodability proof is presented in Appendix E. An example of how the interference alignment scheme works is given in Section V-D.

3) Performance: As we showed in Section V-B, in the first sub-phase we transmit c_t^s bits, with $s = N_{\mathrm{e}}(\mathbf{d})$. In Step $j \in [\min\{s, \mathsf{N}-\mathsf{r}+1, \mathsf{K}-t\}]$ of the second sub-phase, the number of transmitted bits is

$$\frac{\sum_{q=j+1}^{\min\{\mathsf{N}-\mathsf{r}+2,\mathsf{K}-t+1,s\}} \left(\binom{\mathsf{N}-q}{\mathsf{r}-2} - \binom{\mathsf{N}-s}{\mathsf{r}-2}\right)\left(\binom{\mathsf{K}-q}{t-1} - \binom{\mathsf{K}-s}{t-1}\right)}{\binom{\mathsf{N}-1}{\mathsf{r}-1}\binom{\mathsf{K}}{t}} B. \tag{35}$$

Hence, by summing the number of transmitted bits in each step of sub-phase 2 and the number of transmitted bits in sub-phase 1, the load equals $e_t^s + c_t^s$ as defined in (9) and (13), with $s = N_{\mathrm{e}}(\mathbf{d})$.

This concludes the proof of Theorem 3.

D. An Example of Sub-Phase 2 in (11)

We will use the following example to illustrate our interference alignment scheme.

Consider an $(\mathsf{N}, \mathsf{K}, \mathsf{M}, \mathsf{r}) = (5, 10, 1/2, 3)$ shared-link caching problem with correlated files of combinatorial overlaps. There are $\binom{\mathsf{N}}{\mathsf{r}} = 10$ blocks, $W_{\mathcal{S}}$ where $\mathcal{S} \subseteq [5]$ and $|\mathcal{S}| = \mathsf{r} = 3$. The files are

$$F_1 = \{W_{\{1,2,3\}}, W_{\{1,2,4\}}, W_{\{1,2,5\}}, W_{\{1,3,4\}}, W_{\{1,3,5\}}, W_{\{1,4,5\}}\},$$
$$F_2 = \{W_{\{1,2,3\}}, W_{\{1,2,4\}}, W_{\{1,2,5\}}, W_{\{2,3,4\}}, W_{\{2,3,5\}}, W_{\{2,4,5\}}\},$$
$$F_3 = \{W_{\{1,2,3\}}, W_{\{1,3,4\}}, W_{\{1,3,5\}}, W_{\{2,3,4\}}, W_{\{2,3,5\}}, W_{\{3,4,5\}}\},$$
$$F_4 = \{W_{\{1,2,4\}}, W_{\{1,3,4\}}, W_{\{1,4,5\}}, W_{\{2,3,4\}}, W_{\{2,4,5\}}, W_{\{3,4,5\}}\},$$
$$F_5 = \{W_{\{1,2,5\}}, W_{\{1,3,5\}}, W_{\{1,4,5\}}, W_{\{2,3,5\}}, W_{\{2,4,5\}}, W_{\{3,4,5\}}\}.$$

Authorized licensed use limited to: University of Illinois at Chicago Library. Downloaded on November 08,2023 at 23:24:46 UTC from IEEE Xplore. Restrictions apply.

1) Placement Phase: Here $t = \frac{\mathsf{KMr}}{\mathsf{N}} = 3$. We partition each block into $\binom{\mathsf{K}}{t} = 120$ equal-length sub-blocks and denote $W_{\mathcal{S}} = \{W_{\mathcal{S},\mathcal{V}} : \mathcal{V} \subseteq [\mathsf{K}], |\mathcal{V}| = t = 3\}$. Each user $k \in [\mathsf{K}]$ caches $W_{\mathcal{S},\mathcal{V}}$ if $k \in \mathcal{V}$.

2) Delivery Phase: Assume $\mathbf{d} = (1,2,3,4,5,1,2,3,4,5)$, which has $N_{\mathrm{e}}(\mathbf{d}) = 5$ distinct demanded files. We choose as leaders the users in $\mathbf{u} = (1,2,3,4,5)$.

3) First Delivery Sub-Phase: In Step $j \in [\min\{N_{\mathrm{e}}(\mathbf{d}), \mathsf{N}-\mathsf{r}+1, \mathsf{K}-t\}] = [3]$ of the first sub-phase, for each set of users $\mathcal{J} \subseteq [\mathsf{K}] \setminus [j-1]$ where $|\mathcal{J}| = t+1 = 4$ and $j \in \mathcal{J}$, and each set of files $\mathcal{B} \subseteq [\mathsf{N}] \setminus [j]$ where $|\mathcal{B}| = \mathsf{r}-1 = 2$, we transmit $C_{\mathcal{J},\mathcal{B}}$.

At the end of the first sub-phase, as shown in Section V-B, each leader user can recover its desired file.

For the non-leaders, we focus on user 6. By Lemma 1, user 6 can decode $W_{\mathcal{S}_1,\mathcal{V}_1}$ where $1 \in \mathcal{S}_1$, $1 \in \mathcal{V}_1$, and $6 \notin \mathcal{V}_1$, by directly reading off (because in $C_{\mathcal{V}_1\cup\{6\},\mathcal{S}_1\setminus\{1\}}$, user 6 caches all but $W_{\mathcal{S}_1,\mathcal{V}_1}$). Hence, user 6 still needs to recover $W_{\mathcal{S},\mathcal{V}}$ where $1 \in \mathcal{S}$ and $\{1,6\} \cap \mathcal{V} = \emptyset$.

In addition, for each $j \in [g(d_6)+1 : N_{\mathrm{e}}(\mathbf{d})] = [2:5]$, user 6 can recover $W_{\{1\}\cup\mathcal{S}_2,\{u_j\}\cup\mathcal{V}_2}$ where $\mathcal{S}_2 \cap \{d_{u_1},\ldots,d_{u_j}\} = \emptyset$, $\mathcal{V}_2 \cap \{u_1,\ldots,u_j\} = \emptyset$, and $6 \notin \mathcal{V}_2$, by directly reading off from $C_{\{u_j,6\}\cup\mathcal{V}_2,\mathcal{S}_2}$ transmitted in Step j of the first sub-phase (because in $C_{\{u_j,6\}\cup\mathcal{V}_2,\mathcal{S}_2}$, user 6 caches all but $W_{\{1\}\cup\mathcal{S}_2,\{u_j\}\cup\mathcal{V}_2}$).

In order to let each non-leader recover the remaining sub-blocks of its desired file, we proceed the second delivery-phase.

4) Second Delivery Sub-Phase: In Step $j \in [3]$ of the second sub-phase, for each $q \in [j+1:4]$, each $\mathcal{J} \subseteq ([10] \setminus [q-1] \cup \{j\})$ where $|\mathcal{J}| = t+1 = 4$, $\{j,q\} \subseteq \mathcal{J}$, $\mathcal{J} \cap [q+1:5] \neq \emptyset$, and each $\mathcal{B} \subseteq [5]\setminus[q]$ where $|\mathcal{B}| = \mathsf{r}-2 = 1$, we transmit $C_{\mathcal{J},\mathcal{B}}$.

The main objective of the second delivery sub-phase is to let the non-leaders reconstruct the messages in Lemma 4, which are generated by interference alignment. In the following, we focus on the decodability of user 6, and show how user 6 reconstructs the messages in Lemma 4 and how it recovers its desired sub-blocks from those messages.

Observe that leader $g(d_6) = 1$ also demands F_1, we show the decodability of user 6 by induction. For induction step $j' \in [g(d_6)+1 : N_{\mathrm{e}}(\mathbf{d})] = [2:5]$, we prove that user 6 can recover its desired sub-blocks $W_{\mathcal{S},\mathcal{V}}$ where $d_{u_{j'}} \in \mathcal{S}$ or $u_{j'} \in \mathcal{V}$.

We start from $j' = 2$. In the following, we show that user 6 can recover $W_{\{1,2,3\},\mathcal{V}}$ where $\{1,6\} \cap \mathcal{V} = \emptyset$ by interference alignment (i.e., $W_{\{1,2,3\},\{2,3,4\}}$, $W_{\{1,2,3\},\{2,3,5\}}$, $W_{\{1,2,3\},\{2,4,5\}}$, and $W_{\{1,2,3\},\{3,4,5\}}$). We divide the decoding of these 4 sub-blocks of $W_{\{1,2,3\}}$ into three cases:

- We first focus on $W_{\{1,2,3\},\mathcal{V}}$ where $\{1,6\} \cap \mathcal{V} = \emptyset$ and $u_{j'} = 2 \in \mathcal{V}$, e.g., $W_{\{1,2,3\},\{2,3,4\}}$. We will show that user 6 can reconstruct $C_{\{1,2,3,4\},\{1,3\}}$,[9] from which it can recover $W_{\{1,2,3\},\{2,3,4\}}$.
  In Step 1 of the first sub-phase, user 6 receives

$$C_{\{1,2,3,4\},\{2,3\}} = W_{\{1,2,3\},\{2,3,4\}} \oplus W_{\{1,2,3\},\{1,3,4\}} \oplus W_{\{1,2,3\},\{1,2,4\}} \oplus W_{\{2,3,4\},\{1,3,4\}} \oplus W_{\{2,3,4\},\{1,2,4\}} \oplus W_{\{2,3,4\},\{1,2,3\}}; \tag{36}$$

$$C_{\{1,2,3,4\},\{3,4\}} = W_{\{1,3,4\},\{2,3,4\}} \oplus W_{\{1,3,4\},\{1,2,4\}} \oplus W_{\{1,3,4\},\{1,2,3\}} \oplus W_{\{2,3,4\},\{1,3,4\}} \oplus W_{\{2,3,4\},\{1,2,4\}} \oplus W_{\{2,3,4\},\{1,2,3\}}. \tag{37}$$

  By summing (36) and (37), we can obtain

$$C_{\{1,2,3,4\},\{2,3\}} \oplus C_{\{1,2,3,4\},\{3,4\}} = W_{\{1,2,3\},\{2,3,4\}} \oplus W_{\{1,2,3\},\{1,3,4\}} \oplus W_{\{1,2,3\},\{1,2,4\}} \oplus W_{\{1,3,4\},\{2,3,4\}} \oplus W_{\{1,3,4\},\{1,2,4\}} \oplus W_{\{1,3,4\},\{1,2,3\}} \tag{38a}$$

$$= C_{\{1,2,3,4\},\{1,3\}}, \tag{38b}$$

  which shows the property in (34) in Lemma 4. It can be seen by summing (36) and (37), we cancel the interferences from the sub-blocks of $W_{\{2,3,4\}}$ to user 6. From Lemma 1, user 6 can decode $W_{\mathcal{S}_1,\mathcal{V}_1}$ where $1 \in \mathcal{S}_1$ and $1 \in \mathcal{V}_1$. In addition, in

$$C_{\{2,3,4,6\},\{3,4\}} = W_{\{1,3,4\},\{2,4,6\}} \oplus W_{\{1,3,4\},\{2,3,6\}} \oplus W_{\{1,3,4\},\{2,3,4\}} \oplus W_{\{2,3,4\},\{3,4,6\}} \oplus W_{\{2,3,4\},\{2,4,6\}} \oplus W_{\{2,3,4\},\{2,3,6\}}, \tag{39}$$

  which is transmitted in Step 2 of the first sub-phase, user 6 caches all except $W_{\{1,3,4\},\{2,3,4\}}$ such that it can recover $W_{\{1,3,4\},\{2,3,4\}}$ by directly reading off. Hence, user 6 has decoded all except $W_{\{1,2,3\},\{2,3,4\}}$ in (38a) such that it can recover $W_{\{1,2,3\},\{2,3,4\}}$.
  By similar steps, for each desired sub-block $W_{\{1,2,3\},\mathcal{V}}$ where $\{1,6\} \cap \mathcal{V} = \emptyset$ and $u_{j'} = 2 \in \mathcal{V}$, user 6 first reconstructs $C_{\mathcal{V}\cup\{1\},\{1,2,3\}\setminus\{2\}}$ and then recovers $W_{\{1,2,3\},\mathcal{V}}$ from $C_{\mathcal{V}\cup\{1\},\{1,2,3\}\setminus\{2\}}$.
- We then focus on $W_{\{1,2,3\},\mathcal{V}}$ where $\{1,2,6\} \cap \mathcal{V} = \emptyset$, e.g., $W_{\{1,2,3\},\{3,4,5\}}$. We will show that user 6 can reconstruct $C_{\{2,3,4,5\},\{1,3\}}$,[10] from which it can recover $W_{\{1,2,3\},\{3,4,5\}}$.
  In Step 1 of the first sub-phase, user 6 receives

$$C_{\{1,3,4,5\},\{2,3\}} = W_{\{1,2,3\},\{3,4,5\}} \oplus W_{\{1,2,3\},\{1,4,5\}} \oplus W_{\{2,3,4\},\{1,4,5\}} \oplus W_{\{2,3,4\},\{1,3,5\}} \oplus W_{\{2,3,5\},\{1,4,5\}} \oplus W_{\{2,3,5\},\{1,3,4\}}; \tag{40}$$

$$C_{\{1,2,3,5\},\{3,4\}} = W_{\{1,3,4\},\{2,3,5\}} \oplus W_{\{1,3,4\},\{1,2,5\}} \oplus W_{\{2,3,4\},\{1,3,5\}} \oplus W_{\{2,3,4\},\{1,2,5\}} \oplus W_{\{3,4,5\},\{1,2,5\}} \oplus W_{\{3,4,5\},\{1,2,3\}}; \tag{41}$$

$$C_{\{1,2,3,4\},\{3,5\}} = W_{\{1,3,5\},\{2,3,4\}} \oplus W_{\{1,3,5\},\{1,2,4\}} \oplus W_{\{2,3,5\},\{1,3,4\}} \oplus W_{\{2,3,5\},\{1,2,4\}} \oplus W_{\{3,4,5\},\{1,2,4\}} \oplus W_{\{3,4,5\},\{1,2,3\}}. \tag{42}$$

[9]This message is the message $C_{\mathcal{J}\cup\{u_j\},\mathcal{B}\cup\{i\}}$ with $j=1$, $i=1$, $\mathcal{J}=\{2,3,4\}$, $\mathcal{B}=\{3\}$ in Lemma 4. Note that every sub-block in $C_{\{1,2,3,4\},\{1,3\}}$ is desired by user 6.

[10]This message is the message $C_{\mathcal{J}\cup\{u_j\},\mathcal{B}\cup\{i\}}$ with $j=2$, $i=1$, $\mathcal{J}=\{3,4,5\}$, $\mathcal{B}=\{3\}$ in Lemma 4. Note that every sub-block in $C_{\{2,3,4,5\},\{1,3\}}$ is desired by user 6.

Authorized licensed use limited to: University of Illinois at Chicago Library. Downloaded on November 08,2023 at 23:24:46 UTC from IEEE Xplore. Restrictions apply.

In Step 1 of the second sub-phase (with $j=1$, $q=2$, $\mathcal{J}=\{1,2,4,5\}$, $\mathcal{B}=\{3\}$), user 6 receives

$$\begin{aligned}C_{\{1,2,4,5\},\{3\}} &= W_{\{1,2,3\},\{2,4,5\}} \oplus W_{\{1,2,3\},\{1,4,5\}} \\ &\oplus W_{\{1,3,4\},\{2,4,5\}} \oplus W_{\{1,3,4\},\{1,2,5\}} \\ &\oplus W_{\{1,3,5\},\{2,4,5\}} \oplus W_{\{1,3,5\},\{1,2,4\}} \\ &\oplus W_{\{2,3,4\},\{1,4,5\}} \oplus W_{\{2,3,4\},\{1,2,5\}} \\ &\oplus W_{\{2,3,5\},\{1,4,5\}} \oplus W_{\{2,3,5\},\{1,2,4\}} \\ &\oplus W_{\{3,4,5\},\{1,2,5\}} \oplus W_{\{3,4,5\},\{1,2,4\}}. \quad (43)\end{aligned}$$

By summing (40)-(43), we have

$$\begin{aligned}&C_{\{1,3,4,5\},\{2,3\}} \oplus C_{\{1,2,3,5\},\{3,4\}} \oplus C_{\{1,2,3,4\},\{3,5\}} \\ &\oplus C_{\{1,2,4,5\},\{3\}} \\ &= W_{\{1,2,3\},\{3,4,5\}} \oplus W_{\{1,2,3\},\{2,4,5\}} \oplus W_{\{1,3,4\},\{2,4,5\}} \\ &\oplus W_{\{1,3,4\},\{2,3,5\}} \oplus W_{\{1,3,5\},\{2,4,5\}} \oplus W_{\{1,3,5\},\{2,3,4\}} \quad (44a) \\ &= C_{\{2,3,4,5\},\{1,3\}}, \quad (44b)\end{aligned}$$

which shows the property in (33) in Lemma 4. Hence, by (44b), user 6 can reconstruct $C_{\{2,3,4,5\},\{1,3\}}$ while cancelling the interferences in (40)-(43), coinciding with Lemma 4. We then focus on each sub-block in $C_{\{2,3,4,5\},\{1,3\}}$. $W_{\{1,2,3\},\{2,4,5\}}$ can be recovered by user 6 as we showed previously for $W_{\{1,2,3\},\{2,3,4\}}$. For $W_{\{1,3,4\},\{2,4,5\}}$, in

$$\begin{aligned}C_{\{2,4,5,6\},\{3,4\}} &= W_{\{1,3,4\},\{2,5,6\}} \oplus W_{\{1,3,4\},\{2,4,5\}} \\ &\oplus W_{\{2,3,4\},\{4,5,6\}} \oplus W_{\{2,3,4\},\{2,5,6\}} \\ &\oplus W_{\{3,4,5\},\{2,5,6\}} \oplus W_{\{3,4,5\},\{2,4,6\}}, \quad (45)\end{aligned}$$

which is transmitted in Step 2 of the first sub-phase, user 6 caches all except $W_{\{1,3,4\},\{2,4,5\}}$ such that it can recover $W_{\{1,3,4\},\{2,4,5\}}$ by directly reading off. Similarly, user 6 can recover $W_{\{1,3,4\},\{2,3,5\}}$, $W_{\{1,3,5\},\{2,4,5\}}$, and $W_{\{1,3,5\},\{2,3,4\}}$ from Step 2 of the first sub-phase by directly reading off. Hence, in $C_{\{2,3,4,5\},\{1,3\}}$, user 6 has recovered all except $W_{\{1,2,3\},\{3,4,5\}}$ such that user 6 can recover $W_{\{1,2,3\},\{3,4,5\}}$.

- Finally, we consider $W_{\{1,2,3\},\{3,9,10\}}$, where $d_9=4$ and $d_{10}=5$. Notice that, $C_{\{1,2,9,10\},\{3\}}$ is not transmitted in the second sub-phase, because none of users $9,10$ is a leader, which contradicts the constraint on the transmission of the second sub-phase ($\mathcal{J}\cap[q+1:5]\neq\emptyset$ with $q=2$ and $\mathcal{J}=\{1,2,9,10\}$). If user 6 can reconstruct $C_{\{1,2,9,10\},\{3\}}$, by the same decoding procedure as $W_{\{1,2,3\},\{3,4,5\}}$, user 6 can recover $C_{\{2,3,9,10\},\{1,3\}}$,[11] from which it can recover $W_{\{1,2,3\},\{3,9,10\}}$. So in the following, we will prove user 6 can reconstruct $C_{\{1,2,9,10\},\{3\}}$, as described in Lemma 2.
  Notice that $C_{\{1,2,4,10\},\{2,3\}}$ and $C_{\{1,2,4,10\},\{3\}}$ are transmitted in Step 1 of the first and second sub-phases, respectively. Hence, user 6 can obtain

$$\begin{aligned}&C_{\{1,2,4,10\},\{2,3\}} \oplus C_{\{1,2,4,10\},\{3\}} \\ &= W_{\{1,3,4\},\{2,4,10\}} \oplus W_{\{1,3,4\},\{1,2,10\}} \oplus W_{\{1,3,5\},\{2,4,10\}} \\ &\oplus W_{\{1,3,5\},\{1,2,4\}} \oplus W_{\{3,4,5\},\{1,2,10\}} \oplus W_{\{3,4,5\},\{1,2,4\}}. \quad (46)\end{aligned}$$

  On the RHS of (46), $W_{\{1,3,4\},\{2,4,10\}}$ and $W_{\{1,3,5\},\{2,4,10\}}$ can be recovered by user 6 from $C_{\{2,4,6,10\},\{3,4\}}$ and $C_{\{2,4,6,10\},\{3,5\}}$ transmitted in Step 2 of the first sub-phase, respectively (by directly reading off). $W_{\{1,3,4\},\{1,2,10\}}$ and $W_{\{1,3,5\},\{1,2,4\}}$ can be recovered by user 6 because they are cached by user 1 and thus we can use Lemma 1.Item 2. Hence, from (46), user 6 can recover

$$W_{\{3,4,5\},\{1,2,10\}} \oplus W_{\{3,4,5\},\{1,2,4\}}. \quad (47)$$

  Similarly, user 6 can recover

$$W_{\{3,4,5\},\{1,2,5\}} \oplus W_{\{3,4,5\},\{1,2,4\}}, \quad (48)$$
$$W_{\{3,4,5\},\{1,2,5\}} \oplus W_{\{3,4,5\},\{1,2,9\}}, \quad (49)$$

  from $C_{\{1,2,4,5\},\{2,3\}}\oplus C_{\{1,2,4,5\},\{3\}}$ and $C_{\{1,2,8,5\},\{2,3\}}\oplus C_{\{1,2,8,5\},\{3\}}$, respectively. By summing (47)-(49), user 6 can obtain

$$W_{\{3,4,5\},\{1,2,10\}} \oplus W_{\{3,4,5\},\{1,2,9\}}. \quad (50)$$

  Similar to (46), we have

$$\begin{aligned}&C_{\{1,2,9,10\},\{2,3\}} = C_{\{1,2,9,10\},\{3\}} \oplus W_{\{1,3,4\},\{2,9,10\}} \\ &\oplus W_{\{1,3,4\},\{1,2,10\}} \oplus W_{\{1,3,5\},\{2,9,10\}} \oplus W_{\{1,3,5\},\{1,2,9\}} \\ &\oplus W_{\{3,4,5\},\{1,2,10\}} \oplus W_{\{3,4,5\},\{1,2,9\}}. \quad (51)\end{aligned}$$

  On the RHS of (51), $C_{\{1,2,9,10\},\{2,3\}}$ is transmitted in Step 1 of the first sub-phase. In addition, $W_{\{1,3,4\},\{2,9,10\}}$ and $W_{\{1,3,5\},\{2,9,10\}}$ can be recovered by user 6 from $C_{\{2,6,9,10\},\{3,4\}}$ and $C_{\{2,6,9,10\},\{3,5\}}$ transmitted in Step 2 of the first sub-phase, respectively (by directly reading off). $W_{\{1,3,4\},\{1,2,10\}}$ and $W_{\{1,3,5\},\{1,2,9\}}$ can be recovered by user 6 because they are cached by user 1 and thus we can use Lemma 1.Item 2. We also proved in (50) that $W_{\{3,4,5\},\{1,2,10\}} \oplus W_{\{3,4,5\},\{1,2,9\}}$ can be recovered by user 6. Hence, user 6 can reconstruct $C_{\{1,2,9,10\},\{3\}}$ and thus it can recover $W_{\{1,2,3\},\{3,9,10\}}$.
  By similar steps, for each desired sub-block $W_{\{1,2,3\},\mathcal{V}}$ where $\{1,2,6\}\cap\mathcal{V}=\emptyset$, user 6 first reconstructs $C_{\mathcal{V}\cup\{2\},\{1,2,3\}\backslash\{2\}}$ and then recovers $W_{\{1,2,3\},\mathcal{V}}$ from $C_{\mathcal{V}\cup\{2\},\{1,2,3\}\backslash\{2\}}$.

Hence, we proved that user 6 can recover $W_{\{1,2,3\}}$. Similarly, we can prove that user 6 can recover $W_{\mathcal{S}}$ where $\{d_k,d_{u_{j'}}\}=\{1,2\}\subseteq\mathcal{S}$.

For each desired sub-block $W_{\mathcal{S},\mathcal{V}}$ where $d_{u_{j'}}=2\notin\mathcal{S}$, $\{1,6\}\cap\mathcal{V}=\emptyset$, and $u_{j'}=2\in\mathcal{V}$, user 6 can recover $W_{\mathcal{S},\mathcal{V}}$ from $C_{\mathcal{V}\cup\{6\},\mathcal{S}\backslash\{1\}}$ transmitted in Step $u_{j'}=2$ of sub-phase 1 by directly reading off. Hence, we finished the proof of the decodability of user 6 for $j'=2$.

By the induction method, other desired blocks can also be recovered by user 6 with the above decoding procedure. Similarly, the other non-leaders can also recover their desired file from the delivery.

[11] This message is the message $C_{\mathcal{J}\cup\{u_j\},\mathcal{B}\cup\{i\}}$ with $j=2$, $i=1$, $\mathcal{J}=\{3,9,10\}$, $\mathcal{B}=\{3\}$ in Lemma 4. Note that every sub-block in $C_{\{2,3,9,10\},\{1,3\}}$ is desired by user 6.

Authorized licensed use limited to: University of Illinois at Chicago Library. Downloaded on November 08,2023 at 23:24:46 UTC from IEEE Xplore. Restrictions apply.

5) Performance: The achieved load is $41/36 \approx 1.139$ while the converse bound in Theorem 1 is $707/720 \approx 0.982$ and the achieved load in [21] is $7/6 \approx 1.167$.

E. Proof of Theorem 4

For type $\mathcal{D}_s$ where $s \in [\min\{\mathsf{K},\mathsf{N}\}]$ and each corner point $\mathsf{M} = \frac{\mathsf{N}t}{\mathsf{K}\mathsf{r}}$ where $t \in [0:\mathsf{K}]$, from Theorem 3, we can achieve the load in (52g), shown at the bottom of the next page, where (52d), shown at the bottom of the next page, comes from the Pascal's triangle and (52g) comes from the converse bound in Theorem 1. Hence, we proved that the proposed caching scheme in Theorem 3 is order optimal to within a factor of 2 under the constraint of uncoded cache placement for demand type $\mathcal{D}_s$.

Similarly, we can prove that the average load among all possible demands in Theorem 3 is order optimal to within a factor of 2 under the constraint of uncoded cache placement.

F. Proof of Theorem 5

From the decodability proof of the proposed scheme with two delivery sub-phases, we have the following observations, which are proved in Appendix E-C and will help us to further reduce the load for some special cases:

1) Observation 1: when $\mathsf{r} = 2$ or $t = 1$, the transmission of the second sub-phase does not exist and thus each user can recover its desired file from the first sub-phase.
2) Observation 2: for a non-leader k, to decode $W_{\mathcal{S},\mathcal{V}}$ where $d_k \in \mathcal{S}$, $\{d_{u_1},\ldots,d_{u_{g(d_k)-1}}\} \cap \mathcal{S} = \emptyset$ and $\{k,u_1,\ldots,u_{g(d_k)}\} \cap \mathcal{V} = \emptyset$, if there is no user in $\mathcal{V}$ whose demanded file is in $\{d_{u_1},\ldots,d_{u_{g(d_k)-1}}\}$, the multicast messages in the first sub-phase, in Step $g(d_k)$ of the second sub-phase, and in Step $g(d_k)$ in Lemma 2, are enough for user k.
3) Observation 3: for a non-leader k, to decode $W_{\mathcal{S},\mathcal{V}}$ where $d_k \in \mathcal{S}$, $\{d_{u_1},\ldots,d_{u_{g(d_k)-1}}\} \cap \mathcal{S} = \emptyset$, $\{k,u_1,\ldots,u_{g(d_k)}\} \cap \mathcal{V} = \emptyset$, and $(\cup_{k'\in\mathcal{V}}\{d_{k'}\}) \cap (\mathcal{S} \setminus \{d_k\}) = \emptyset$, user k only needs the transmission of the first sub-phase.

In the following, we will show if $\mathsf{r} \in \{1,2,\mathsf{N}-1,\mathsf{N}\}$ or $t \in \{0,1,2,\mathsf{K}-1,\mathsf{K}\}$ or $s \in [\min\{\mathsf{K},\mathsf{N},4\}]$, the transmission of the second sub-phase is not needed. Notice that the load of the first sub-phase (i.e., c_t^s) coincides with the proposed converse bound in Theorem 1 (i.e., $\mathsf{R}_{\mathrm{u}}^{\star}(\mathsf{M},\mathsf{s})$). Hence, for the above cases, the transmission of the first sub-phase is optimal under the constraint of uncoded cache placement.

When $\mathsf{r} \in \{1,\mathsf{N}\}$, the considered problem is equivalent to the MAN caching problem, the first sub-phase is equivalent to the caching scheme in [4], which is optimal under the constraint of uncoded cache placement.

When $t \in \{0,\mathsf{K}\}$, it is simple to achieve the optimality by transmitting all demanded files or nothing.

When $\mathsf{r} = 2$ or $t = 1$, as shown in Observation 1, each non-leader can recover its desired files from the transmission of the first sub-phase.

When $t = \mathsf{K}-1$, there is only one step in the first sub-phase. By Lemma 1.Item 2, it can be seen that any non-leader can recover its desired blocks from Step 1 of the first sub-phase. Hence, the second sub-phase is not necessary.

We now consider $\mathsf{r} = \mathsf{N}-1$ or $t = 2$ and let each non-leader k recover $W_{\mathcal{S},\mathcal{V}}$ where $d_k \in \mathcal{S}$, $\{d_{u_1},\ldots,d_{u_{g(d_k)-1}}\} \cap \mathcal{S} = \emptyset$ and $\{k,u_1,\ldots,u_{g(d_k)}\} \cap \mathcal{V} = \emptyset$, by the transmission of the first sub-phase. The fact that the first sub-phase is enough for these two cases, is because user k can reconstruct its required the multicast messages transmitted in the second sub-phase, from the first sub-phase. The detail of the decodability proof for these two cases could be found in Appendix F.

In conclusion, for the cases where $\mathsf{r} \in \{1,2,\mathsf{N}-1,\mathsf{N}\}$ or $t \in \{1,2,\mathsf{K}-1,\mathsf{K}\}$, we proved that from the first delivery sub-phase, each user can recover its desired file. Comparing the converse bound in Theorem 1 and the achieved load (given in Section V-F), we have the optimality for Theorem 5.Case 1 where $\mathsf{r} \in \{1,2,\mathsf{N}-1,\mathsf{N}\}$. The optimality for Theorem 5.Case 2 where either $\mathsf{K}\mathsf{r}\mathsf{M}/\mathsf{N} \le 2$ or $\mathsf{K}\mathsf{r}\mathsf{M}/\mathsf{N} \ge \mathsf{K}-1$, is due to the fact that in the converse bound (10), $c_t^{N_{\mathrm{e}}(\mathbf{d})}$ is convex in terms of t and when $t \in \{0,1,2,\mathsf{K}-1,\mathsf{K}\}$, our proposed scheme is optimal.

Finally, we will prove the optimality of $\mathsf{R}_{\mathrm{u}}^{\star}(\mathsf{M},s)$ for Theorem 5.Case 3 where $s \in [\min\{\mathsf{K},\mathsf{N},4\}]$. We consider the following two cases.

1) $\min\{\mathsf{K},\mathsf{N}\} \le 4$. Theorem 5.Case 1 covers all possible values of r when $3 \ge \mathsf{N}-1$, and Theorem 5.Case 2 covers all possible values of M when $3 \ge \mathsf{K}-1$. Hence, when $\min\{\mathsf{K},\mathsf{N}\} \le 4$, we can prove the optimality.
2) $\min\{\mathsf{K},\mathsf{N}\} > 4$. In this case, $s = |\mathcal{N}_{\mathbf{d}}([\mathsf{K}])| \le 4$. For each subset of files $\mathcal{T} \subseteq [\mathsf{N}] \setminus \mathcal{N}_{\mathbf{d}}([\mathsf{K}])$ where $\mathsf{r}-4 \le |\mathcal{T}| < \mathsf{r}$, we can gather all blocks $W_{\mathcal{S}}$ where $\mathcal{S} \subseteq [\mathsf{N}]$, $|\mathcal{S}| = \mathsf{r}$, $\mathcal{S} \setminus \mathcal{N}_{\mathbf{d}}([\mathsf{K}]) = \mathcal{T}$. The proposed first delivery sub-phase on these blocks is equivalent to the first delivery sub-phase for $\mathcal{N}'_{\mathbf{d}}([\mathsf{K}]) = \mathsf{N}' = s$, $\mathsf{K}' = \mathsf{K}$, $\mathsf{r}' = \mathsf{r}-|\mathcal{T}|$, and $t' = t$. Since we proved the decodability of the proposed first delivery sub-phase for the system including up to 4 files, we can prove the blocks in this group can be recovered by the demanding users. Hence, we prove that each user can recover its desired file from the first delivery sub-phase.

As a result, we proved that when $s \in [\min\{\mathsf{K},\mathsf{N},4\}]$, each user can recover its desired file from the first delivery sub-phase, and thus we proved the optimality for Theorem 5.Case 3.

VI. Conclusion

In this paper, we studied the coded caching problem with correlated files of combinatorial overlaps and aimed to minimize the average load over the uniform demand distribution. We proposed a converse bound under the constraint of uncoded cache placement and a new coded caching scheme based on interference alignment, containing two sub-phases. For any demand type, under the constraint of uncoded cache placement, our caching scheme is optimal to within a factor of 2. For the demand type $\mathcal{D}_s$ where $s = \mathsf{K}$ or $s \in [\min\{\mathsf{K},\mathsf{N},4\}]$, or for the case with any demand type with $\mathsf{r} \in \{1,2,\mathsf{N}-1,\mathsf{N}\}$ or $\mathsf{K}\mathsf{r}\mathsf{M} \le 2\mathsf{N}$ or $\mathsf{K}\mathsf{r}\mathsf{M} \ge (\mathsf{K}-1)\mathsf{N}$, the first sub-phase of the proposed scheme is decodable and optimal under the constraint of uncoded cache placement. As an extension, the above exact and order optimal results can be extended to the worst-case loads. As by-products, we showed that the proposed strategy

Authorized licensed use limited to: University of Illinois at Chicago Library. Downloaded on November 08,2023 at 23:24:46 UTC from IEEE Xplore. Restrictions apply.

which jointly serves the users' demands reduces the load of existing schemes for the coded caching problem with multiple requests; the proposed scheme for distinct demands can be also extended to the coded distributed computing problem with a central server, which achieves the optimal transmission load over the binary field.

APPENDIX A
PROOF OF LEMMA 1

For a given demand vector $\mathbf{d}$, let $s = N_{\mathrm{e}}(\mathbf{d})$, $j_{\max} = \min\{s, \mathsf{N}-\mathsf{r}+1, \mathsf{K}-t\}$, and order the leader users as $(u_1, \ldots, u_s)$. Recall that in Step $j \in [j_{\max}]$ of delivery sub-phase 1 of the scheme in (11) we satisfy the demand of leader user u_j as follows: for each set of users $\mathcal{J} \subseteq [\mathsf{K}] \setminus \{u_1, \ldots, u_{j-1}\}$ such that $|\mathcal{J}| = t+1$ and $u_j \in \mathcal{J}$, and for each set of files $\mathcal{B} \subseteq [\mathsf{N}] \setminus \{d_{u_1}, \ldots, d_{u_j}\}$ such that $|\mathcal{B}| = \mathsf{r}-1$, we transmit the multicast message in (28), which we re-write as

$$C_{\mathcal{J},\mathcal{B}} = \bigoplus_{k\in\mathcal{J}:d_k\notin\mathcal{B}} W_{\mathcal{B}\cup\{d_k\},\mathcal{J}\setminus\{k\}} \tag{53a}$$

$$\oplus \bigoplus_{k\in\mathcal{J}:d_k\in\mathcal{B}} \Big(W_{\mathcal{B}\cup\{d_{u_j}\},\mathcal{J}\setminus\{k\}} \oplus \bigoplus_{i\in\mathcal{N}_{\mathbf{d}}(\mathcal{J})\setminus(\mathcal{B}\cup\{d_{u_j}\})} W_{\mathcal{B}\cup\{i\},\mathcal{J}\setminus\{k\}} \Big). \tag{53b}$$

By construction (i.e., $d_{u_j} \notin \mathcal{B}$), $C_{\mathcal{J},\mathcal{B}}$ in (53) contains only one sub-block desired by user u_j (which is $W_{\mathcal{B}\cup\{d_{u_j}\},\mathcal{J}\setminus\{u_j\}}$), while all other sub-blocks in $C_{\mathcal{J},\mathcal{B}}$ are in its cache. Based on this observation, we introduce the following terminology:

Directly read off. The observation made for leader user u_j actually holds for every user $k \in \mathcal{J}$ where $d_k \notin \mathcal{B}$ (i.e., term in (53a)). Thus, we say that user k 'directly reads off' its desired sub-block $W_{\mathcal{B}\cup\{d_k\},\mathcal{J}\setminus\{k\}}$ from the multicast message $C_{\mathcal{J},\mathcal{B}}$.

Indirectly read off. For user $k \in \mathcal{J}$ where $d_k \in \mathcal{B}$, its desired sub-blocks appear in $C_{\mathcal{J},\mathcal{B}}$ as the linear combination $W_{\mathcal{B}\cup\{d_{u_j}\},\mathcal{J}\setminus\{k\}} + \bigoplus_{i\in\mathcal{N}_{\mathbf{d}}(\mathcal{J})\setminus(\mathcal{B}\cup\{d_{u_j}\})} W_{\mathcal{B}\cup\{i\},\mathcal{J}\setminus\{k\}}$ (i.e., term in (53b)), while the other sub-blocks appearing in $C_{\mathcal{J},\mathcal{B}}$ are cached by user k. We will prove later that user k can recover $W_{\mathcal{B}\cup\{i\},\mathcal{J}\setminus\{k\}}$ where $i \in \mathcal{N}_{\mathbf{d}}(\mathcal{J})\setminus(\mathcal{B}\cup\{d_{u_j}\})$ from other multicast messages. Thus user k can 'indirectly read off' its desired sub-block $W_{\mathcal{B}\cup\{d_{u_j}\},\mathcal{J}\setminus\{k\}}$ from the multicast message $C_{\mathcal{J},\mathcal{B}}$.

Lemma 1 is proved by induction.

A. Step 1

Lemma 1.Item 1: We focus on one set of users $\mathcal{J} \subseteq [\mathsf{K}]$ where $|\mathcal{J}| = t+1$ and $u_1 \in \mathcal{J}$, and one set of files $\mathcal{B} \subseteq [\mathsf{N}] \setminus \{d_{u_1}\}$ where $|\mathcal{B}| = \mathsf{r}-1$. We will prove that from Step 1, each user in $k \in \mathcal{J}$ can recover all sub-blocks in $C_{\mathcal{J},\mathcal{B}}$. We consider two cases:

- $d_k \notin \mathcal{B}$: in $C_{\mathcal{J},\mathcal{B}}$ user k caches all sub-blocks except $W_{\mathcal{B}\cup\{d_k\},\mathcal{J}\setminus\{k\}}$. Hence, user k can recover $W_{\mathcal{B}\cup\{d_k\},\mathcal{J}\setminus\{k\}}$ by directly reading off.
- $d_k \in \mathcal{B}$: in $C_{\mathcal{J},\mathcal{B}}$ user k caches all sub-blocks except $W_{\mathcal{B}\cup\{i\},\mathcal{J}\setminus\{k\}}$, where $i \in \mathcal{N}_{\mathbf{d}}(\mathcal{J}) \setminus \mathcal{B}$.
  - If $i \neq d_{u_1}$, user k can recover $W_{\mathcal{B}\cup\{i\},\mathcal{J}\setminus\{k\}}$ from $C_{\mathcal{J},(\mathcal{B}\cup\{i\})\setminus\{d_k\}}$ by directly reading off as the similar reason described in the above case.
  - If $i = d_{u_1}$, since we proved that user k can recover all sub-blocks in $C_{\mathcal{J},\mathcal{B}}$ except $W_{\mathcal{B}\cup\{d_{u_1}\},\mathcal{J}\setminus\{k\}}$, then it can be seen that user k can recover $W_{\mathcal{B}\cup\{d_{u_1}\},\mathcal{J}\setminus\{k\}}$ by indirectly reading off.

In conclusion, user k can recover all sub-blocks in $C_{\mathcal{J},\mathcal{B}}$. Hence, we proved Lemma 1.Item 1 for Step 1.

Lemma 1.Item 2: Note that user u_1 can recover $W_{\mathcal{S},\mathcal{V}}$ where $d_{u_1} \in \mathcal{S}$ and $u_1 \in \mathcal{V}$, from its cache. Hence, in the following, we will prove that any user $k \in ([\mathsf{K}] \setminus \{u_1\})$ can recover each $W_{\mathcal{S},\mathcal{V}}$ where $d_k \in \mathcal{S}$, $u_1 \in \mathcal{V}$ and $k \notin \mathcal{V}$, from Step 1. We consider two cases:

$$\begin{aligned}
&c_t^s + e_t^s \\
&= \sum_{j\in[\min\{s,\mathsf{N}-\mathsf{r}+1,\mathsf{K}-t\}]} \frac{\binom{\mathsf{N}-j}{\mathsf{r}-1}\binom{\mathsf{K}-j}{t} + \sum_{q=j+1}^{\min\{\mathsf{N}-\mathsf{r}+2,\mathsf{K}-t+1,s\}} \left(\binom{\mathsf{N}-q}{\mathsf{r}-2} - \binom{\mathsf{N}-s}{\mathsf{r}-2}\right)\left(\binom{\mathsf{K}-q}{t-1} - \binom{\mathsf{K}-s}{t-1}\right)}{\binom{\mathsf{N}-1}{\mathsf{r}-1}\binom{\mathsf{K}}{t}} && \text{(52a)}\\
&\le \sum_{j\in[\min\{s,\mathsf{N}-\mathsf{r}+1,\mathsf{K}-t\}]} \frac{\binom{\mathsf{N}-j}{\mathsf{r}-1}\binom{\mathsf{K}-j}{t} + \sum_{q=j+1}^{\min\{\mathsf{N}-\mathsf{r}+2,\mathsf{K}-t+1,s\}} \binom{\mathsf{N}-q}{\mathsf{r}-2}\binom{\mathsf{K}-q}{t-1}}{\binom{\mathsf{N}-1}{\mathsf{r}-1}\binom{\mathsf{K}}{t}} && \text{(52b)}\\
&\le \sum_{j\in[\min\{s,\mathsf{N}-\mathsf{r}+1,\mathsf{K}-t\}]} \frac{\binom{\mathsf{N}-j}{\mathsf{r}-1}\binom{\mathsf{K}-j}{t} + \sum_{q=j+1}^{\min\{\mathsf{N}-\mathsf{r}+2,\mathsf{K}-t+1,s\}} \binom{\mathsf{N}-q}{\mathsf{r}-2}\binom{\mathsf{K}-j-1}{t-1}}{\binom{\mathsf{N}-1}{\mathsf{r}-1}\binom{\mathsf{K}}{t}} && \text{(52c)}\\
&\le \sum_{j\in[\min\{s,\mathsf{N}-\mathsf{r}+1,\mathsf{K}-t\}]} \frac{\binom{\mathsf{N}-j}{\mathsf{r}-1}\binom{\mathsf{K}-j}{t} + \binom{\mathsf{N}-j}{\mathsf{r}-1}\binom{\mathsf{K}-j-1}{t-1}}{\binom{\mathsf{N}-1}{\mathsf{r}-1}\binom{\mathsf{K}}{t}} && \text{(52d)}\\
&\le 2 \sum_{j\in[\min\{s,\mathsf{N}-\mathsf{r}+1,\mathsf{K}-t\}]} \frac{\binom{\mathsf{N}-j}{\mathsf{r}-1}\binom{\mathsf{K}-j}{t}}{\binom{\mathsf{N}-1}{\mathsf{r}-1}\binom{\mathsf{K}}{t}} && \text{(52e)}\\
&= 2c_t^s && \text{(52f)}\\
&\le 2\mathsf{R}_{\mathrm{u}}^{\star}(\mathsf{M},\mathsf{s}), && \text{(52g)}
\end{aligned}$$

Authorized licensed use limited to: University of Illinois at Chicago Library. Downloaded on November 08,2023 at 23:24:46 UTC from IEEE Xplore. Restrictions apply.

- $d_{u_1} \notin \mathcal{S}$. We can see that $W_{\mathcal{S},\mathcal{V}}$ appears in $C_{\mathcal{V}\cup\{k\},\mathcal{S}\setminus\{d_k\}}$. By Lemma 1.Item 1 for Step 1, we prove that user k can recover $W_{\mathcal{S},\mathcal{V}}$.
- $d_{u_1} \in \mathcal{S}$. We can see that $W_{\mathcal{S},\mathcal{V}}$ appears in $C_{\mathcal{V}\cup\{k\},\mathcal{S}\setminus\{d_{u_1}\}}$. By Lemma 1.Item 1 for Step 1, we prove that user k can recover $W_{\mathcal{S},\mathcal{V}}$.

Hence, we proved Lemma 1.Item 2 for Step 1.

Lemma 1.Item 3: We then focus on one user k whose demanded file is in $[\mathsf{N}]\setminus\{d_{u_1}\}$, and one sub-block $W_{\mathcal{S},\mathcal{V}}$ where $\{d_k, d_{u_1}\} \subseteq \mathcal{S}$ and $\{u_1, k\} \cap \mathcal{V} = \emptyset$. In $C_{\mathcal{V}\cup\{u_1\},\mathcal{S}\setminus\{d_{u_1}\}}$, all sub-blocks are desired by user k while only one of them is desired by user u_1 (which is $W_{\mathcal{S},\mathcal{V}}$) and the others are cached by user u_1. From Lemma 1.Item 2 for Step 1, user k has recovered all desired sub-blocks which are cached by user u_1, and thus user k can recover $W_{\mathcal{S},\mathcal{V}}$ from $C_{\mathcal{V}\cup\{u_1\},\mathcal{S}\setminus\{d_{u_1}\}}$. Hence, we proved Lemma 1.Item 3 for Step 1.

In summary, we proved Lemma 1 for Step 1.

B. Step j

We focus one $j \in [\min\{N_{\mathrm{e}}(\mathbf{d}), \mathsf{N}-\mathsf{r}+1, \mathsf{K}-t\}]$ and assume that Lemma 1 holds for the first $j-1$ steps. In the following, we prove that Lemma 1 holds for Step j.

Lemma 1.Item 1: We focus on one set of users $\mathcal{J} \subseteq ([\mathsf{K}] \setminus \{u_1, \ldots, u_{j-1}\})$ where $|\mathcal{J}| = t+1$ and $u_j \in \mathcal{J}$, and one set of files $\mathcal{B} \subseteq ([\mathsf{N}] \setminus \{d_{u_1}, \ldots . d_{u_j}\})$ where $|\mathcal{B}| = \mathsf{r}-1$. We will prove that from the transmission until Step j, each user in $k \in \mathcal{J}$ can recover all sub-blocks in $C_{\mathcal{J},\mathcal{B}}$. We consider two cases:

- $d_k \notin \mathcal{B}$. In this case, in $C_{\mathcal{J},\mathcal{B}}$ user k caches all sub-blocks except $W_{\mathcal{B}\cup\{d_k\},\mathcal{J}\setminus\{k\}}$. Hence, user k can recover $W_{\mathcal{B}\cup\{d_k\},\mathcal{J}\setminus\{k\}}$ by directly reading off.
- $d_k \in \mathcal{B}$. In this case, $d_k \notin \{d_{u_1}, \ldots, d_{u_j}\}$. In $C_{\mathcal{J},\mathcal{B}}$ user k caches all sub-blocks except $W_{\mathcal{B}\cup\{i\},\mathcal{J}\setminus\{k\}}$, where $i \in \mathcal{N}_{\mathbf{d}}(\mathcal{J}) \setminus \mathcal{B}$.
  - If $i \in \{d_{u_1}, \ldots, d_{u_{j-1}}\}$, by the induction assumption, user k has already recovered the whole block $W_{\mathcal{B}\cup\{i\}}$.
  - If $i \notin \{d_{u_1}, \ldots, d_{u_j}\}$, user k can recover $W_{\mathcal{B}\cup\{i\},\mathcal{J}\setminus\{k\}}$ from $C_{\mathcal{J},(\mathcal{B}\cup\{i\})\setminus\{d_k\}}$ transmitted in Step j by directly reading off.
  - If $i = d_{u_j}$, in $C_{\mathcal{J},\mathcal{B}}$ user k has cached or recovered all sub-blocks except $W_{\mathcal{B}\cup\{d_{u_j}\},\mathcal{J}\setminus\{k\}}$. Hence, user k can recover $W_{\mathcal{B}\cup\{d_{u_j}\},\mathcal{J}\setminus\{k\}}$ by indirectly reading off.

In conclusion, user k can recover all sub-blocks in $C_{\mathcal{J},\mathcal{B}}$, and thus we proved Lemma 1.Item 1 for Step j.

Lemma 1.Item 2: Note that user u_j can recover $W_{\mathcal{S},\mathcal{V}}$ where $d_k \in \mathcal{S}$ and $u_j \in \mathcal{V}$, from its cache. Hence, in the following, we will prove any user $k \in ([\mathsf{K}] \setminus \{u_j\})$ where $d_k \notin \{d_{u_1}, \ldots, d_{u_{j-1}}\}$, can recover each $W_{\mathcal{S},\mathcal{V}}$ where $d_k \in \mathcal{S}$, $\{d_{u_1}, \ldots, d_{u_{j-1}}\} \cap \mathcal{S} = \emptyset$, $u_j \in \mathcal{V}$, and $\{k, u_1, \ldots, u_{j-1}\} \cap \mathcal{V} = \emptyset$, at the end of Step j. We consider two cases:

- $d_{u_j} \notin \mathcal{S}$. We can see that $W_{\mathcal{S},\mathcal{V}}$ appears in $C_{\mathcal{V}\cup\{k\},\mathcal{S}\setminus\{d_k\}}$ transmitted in Step j. By Lemma 1.Item 1 for Step j, we prove that user k can recover $W_{\mathcal{S},\mathcal{V}}$.
- $d_{u_j} \in \mathcal{S}$. We can see that $W_{\mathcal{S},\mathcal{V}}$ appears in $C_{\mathcal{V}\cup\{k\},\mathcal{S}\setminus\{d_{u_j}\}}$ transmitted in Step j. By Lemma 1.Item 1 for Step j, we prove that user k can recover $W_{\mathcal{S},\mathcal{V}}$.

Hence, we proved Lemma 1.Item 2 for Step j.

Lemma 1.Item 3: We then focus on one user k where $d_k \in ([\mathsf{N}] \setminus \{d_{u_1}, \ldots, d_{u_j}\})$, and one sub-block $W_{\mathcal{S},\mathcal{V}}$ where $\{d_k, d_{u_j}\} \subseteq \mathcal{S}$, $\{d_{u_1}, \ldots, d_{u_{j-1}}\} \cap \mathcal{S} = \emptyset$, and $\{k, u_1, \ldots, u_j\} \cap \mathcal{V} = \emptyset$. In $C_{\mathcal{V}\cup\{u_j\},\mathcal{S}\setminus\{d_{u_j}\}}$ transmitted in Step j, all sub-blocks are desired by user k, while only one of them is desired by user u_j (which is $W_{\mathcal{S},\mathcal{V}}$) and the others are cached by user u_j. From Lemma 1.Item 2 for Step j, user k has recovered all desired sub-blocks which are cached by user u_j, and thus user k can recover $W_{\mathcal{S},\mathcal{V}}$ from $C_{\mathcal{V}\cup\{u_j\},\mathcal{S}\setminus\{d_{u_j}\}}$. Hence, we proved Lemma 1.Item 3 for Step j.

In conclusion, we proved Lemma 1.

Appendix B
Proof of Lemma 2

Focus on Step $j \in [\min\{N_{\mathrm{e}}(\mathbf{d}), \mathsf{N}-\mathsf{r}+1, \mathsf{K}-t\}]$ and $q \in [j+1 : \min\{\mathsf{N}-\mathsf{r}+2, \mathsf{K}-t+1, N_{\mathrm{e}}(\mathbf{d})\}]$. We will prove that each user k with $d_k = d_{u_j}$ can recover $C_{\mathcal{J},\mathcal{B}}$, for each $\mathcal{J} \subseteq [\mathsf{K}] \setminus (\mathcal{L}(\mathbf{d}) \setminus \{u_j, u_q\})$ where $|\mathcal{J}| = t+1$ and $\{u_j, u_q\} \subseteq \mathcal{J}$, and each $\mathcal{B} \subseteq ([\mathsf{N}] \setminus \{d_{u_1}, \ldots, d_{u_q}\})$ where $|\mathcal{B}| = \mathsf{r}-2$ and $\mathcal{B} \cap \mathcal{N}_{\mathbf{d}}([\mathsf{K}]) \neq \emptyset$, at the end of Step j of sub-phase 2.

Let $\mathcal{J}' = \mathcal{J} \setminus \{u_j, u_q\}$. It can be seen that $\mathcal{J}'$ only contains non-leaders. Recall that $\mathcal{N}_{\mathbf{d}}(\mathcal{T})$ is the union set of files demanded by the users in $\mathcal{T}$, and $\mathcal{L}_{\mathbf{d}}(\mathcal{T})$ is the union set of leader users demanding the files in $\mathcal{N}_{\mathbf{d}}(\mathcal{T})$. Given $\mathcal{J}'$, we define a family of sets $\mathscr{F}(\mathcal{J}')$ containing each set $\mathcal{J}' \cup (\mathcal{L}_{\mathbf{d}}(\mathcal{J}')\setminus\{u_j, u_q\})\setminus\mathcal{F}$, where $\mathcal{F} \subseteq \mathcal{J}' \cup (\mathcal{L}_{\mathbf{d}}(\mathcal{J}')\setminus\{u_j, u_q\})$, $|\mathcal{F}| = |\mathcal{L}_{\mathbf{d}}(\mathcal{J}')\setminus\{u_j, u_q\}|$ and $\mathcal{N}_{\mathbf{d}}(\mathcal{F}) = \mathcal{N}_{\mathbf{d}}(\mathcal{J}')\setminus\{d_{u_j}, d_{u_q}\}$. In plain words, for each file in $\mathcal{N}_{\mathbf{d}}(\mathcal{J}')\setminus\{d_{u_j}, d_{u_q}\}$, we replace one or zero user in $\mathcal{J}'$ demanding this file by the leader user demanding this file; the resulting set is a set in $\mathscr{F}(\mathcal{J}')$. For example, $\mathcal{J}' = \{5,6,7,8\}$ where $d_{u_j} = 1$, $d_{u_q} = 2$, $d_5 = d_6 = 3$, $d_7 = 4$, and $d_8 = 2$. Assume that the leader user demanding F_3 is user 3 and the leader user demanding F_4 is user 4. After replacing users $5, 7$ in $\mathcal{J}'$ by users $3, 4$, we obtain the set $\{3,4,6,8\} \in \mathscr{F}(\mathcal{J}')$. Similarly, in this example we have

$$\mathscr{F}(\mathcal{J}') = \big\{\{3,4,6,8\}, \{3,4,5,8\}, \{3,6,7,8\}, \{3,5,7,8\}, \{4,5,6,8\}, \{5,6,7,8\}\big\}.$$

For each $\mathcal{J}_1 \in \mathscr{F}(\mathcal{J}')$, with a slight abuse of notation, we let

$$Q_{\mathcal{J}_1} := \bigoplus_{k' \in \mathcal{J}_1} \bigoplus_{\substack{\mathcal{S} \subseteq (\mathcal{N}_{\mathbf{d}}(\mathcal{J}_1) \cup \mathcal{B} \setminus \{d_{u_j}, d_{u_q}\}):\\ \mathcal{B} \subseteq \mathcal{S}, d_{k'} \in \mathcal{S}}} W_{\mathcal{S},\mathcal{J}_1\cup\{u_j,u_q\}\setminus\{k'\}}. \tag{54}$$

In plain words, $Q_{\mathcal{J}_1}$ is obtained by removing all sub-blocks in the blocks desired by user u_j or u_q from $C_{\mathcal{J}_1\cup\{u_j,u_q\},\mathcal{B}}$.

Authorized licensed use limited to: University of Illinois at Chicago Library. Downloaded on November 08,2023 at 23:24:46 UTC from IEEE Xplore. Restrictions apply.

By definition, we have

$$C_{\mathcal{J}_1\cup\{u_j,u_q\},\mathcal{B}} \oplus C_{\mathcal{J}_1\cup\{u_j,u_q\},\mathcal{B}\cup\{u_q\}} \oplus \mathcal{Q}_{\mathcal{J}_1}$$
$$= \left(\bigoplus_{k'\in\mathcal{J}_1\cup\{u_j\}} \bigoplus_{\substack{\mathcal{S}\subseteq(\mathcal{N}_{\mathbf{d}}(\mathcal{J}_1\cup\{u_j\})\cup\mathcal{B})\setminus\{d_{u_q}\}:\\ \mathcal{B}\subseteq\mathcal{S},d_{k'}\in\mathcal{S}}} W_{\mathcal{S},\mathcal{J}_1\cup\{u_j,u_q\}\setminus\{k'\}} \right) \oplus \mathcal{Q}_{\mathcal{J}_1} \tag{55a}$$
$$= \bigoplus_{k'\in\mathcal{J}_1\cup\{u_j\}} \bigoplus_{\substack{\mathcal{S}\subseteq(\mathcal{N}_{\mathbf{d}}(\mathcal{J}_1\cup\{u_j\})\cup\mathcal{B})\setminus\{d_{u_q}\}:\\ \mathcal{B}\subseteq\mathcal{S},\{d_{k'},d_{u_j}\}\subseteq\mathcal{S}}} W_{\mathcal{S},\mathcal{J}_1\cup\{u_j,u_q\}\setminus\{k'\}}. \tag{55b}$$

On the RHS of (55b), if $k'\neq u_j$, $W_{\mathcal{S},\mathcal{J}_1\cup\{u_j,u_q\}\setminus\{k'\}}$ is cached by u_j and by Lemma 1.Item 2, user k can recover $W_{\mathcal{S},\mathcal{J}_1\cup\{u_j,u_q\}\setminus\{k'\}}$. We then consider $k'=u_j$ and focus on $W_{\mathcal{S},\mathcal{J}_1\cup\{u_j,u_q\}\setminus\{k'\}}=W_{\mathcal{S},\mathcal{J}_1\cup\{u_q\}}$. Since $u_q\notin\mathcal{S}$, it will be proved later in Remark 6 that $W_{\mathcal{S},\mathcal{J}_1\cup\{u_q\}}$ can be recovered by user k at the end of Step $g(d_k)=j$ of sub-phase 2. Hence, user k can reconstruct the RHS of (55b) at the end of Step $g(d_k)=j$ of sub-phase 2.

In addition, for each $\mathcal{J}_2\in\mathscr{F}(\mathcal{J}')$ where $\mathcal{J}_2\neq\mathcal{J}'$, since there exists at least one leader in $\mathcal{J}_2$, it can be seen that $C_{\mathcal{J}_2\cup\{u_j,u_q\},\mathcal{B}\cup\{u_q\}}$ and $C_{\mathcal{J}_2\cup\{u_j,u_q\},\mathcal{B}}$ are received in (or before) Step j of the first and second sub-phases, respectively. Hence, user k can reconstruct $\mathcal{Q}_{\mathcal{J}_2}$ from (55b) at the end of Step j of sub-phase 2.

At the end of this proof, we will prove the following equation.

$$\bigoplus_{\mathcal{J}_1\in\mathscr{F}(\mathcal{J}')} Q_{\mathcal{J}_1}=0. \tag{56}$$

In (56), all the messages except $\mathcal{Q}_{\mathcal{J}'}$ are recovered by user k such that each user can reconstruct $\mathcal{Q}_{\mathcal{J}'}$. In addition, $C_{\mathcal{J}'\cup\{u_j,u_q\},\mathcal{B}\cup\{u_q\}}=C_{\mathcal{J},\mathcal{B}\cup\{u_q\}}$ is transmitted in Step j of the first sub-phase. Hence, from (55b), user k can reconstruct $C_{\mathcal{J}'\cup\{u_j,u_q\},\mathcal{B}}=C_{\mathcal{J},\mathcal{B}}$ at the end of Step j of sub-phase 2.

Finally, we will prove (56). We focus on any sub-block $W_{\mathcal{S},\mathcal{V}}$ in (56) and assume that $W_{\mathcal{S},\mathcal{V}}$ is in $\mathcal{Q}_{\mathcal{J}_1}$, which is desired by user k_1. Hence, $k_1\in\mathcal{J}_1$, $\mathcal{V}=\mathcal{J}_1\cup\{u_j,u_q\}\setminus\{k_1\}$, and $d_{k_1}\notin\{d_{u_j},d_{u_q}\}$. By the construction of $\mathscr{F}(\mathcal{J}')$, there exists only one user in $\mathcal{J}'\cup\{u_{g(d_{k_1})}\}$ demanding d_{k_1}, who is not in $\mathcal{J}_1$. We assume that this user is k_2. It can be seen that $W_{\mathcal{S},\mathcal{V}}$ desired by k_2, is also in $\mathcal{Q}_{\mathcal{J}_2}$ where $\mathcal{J}_2=\mathcal{J}_1\cup\{k_2\}\setminus\{k_1\}$. In addition, except $\mathcal{J}_1$ and $\mathcal{J}_2$, there does not exist other $\mathcal{J}_3\in\mathscr{F}(\mathcal{J}')$ where $\mathcal{Q}_{\mathcal{J}_3}$ contains $W_{\mathcal{S},\mathcal{V}}$, because $\mathcal{V}\setminus\{d_{u_j},d_{u_q}\}=\mathcal{J}_1\setminus\{k_1\}$ cannot be a subset of $\mathcal{J}_3$. Hence, $W_{\mathcal{S},\mathcal{V}}$ appears twice in (56) and we proved (56).

Appendix C
Proof of Lemma 3

A. Proof of (33)

Focus on one $\mathcal{J}\subseteq[\mathsf{K}]$ where $|\mathcal{J}|=t+1$, and one $\mathcal{B}\subseteq[\mathsf{N}]$ where $|\mathcal{B}|=\mathsf{r}-1$. To prove (33), it is equivalent to prove that

$$\bigoplus_{k\in\mathcal{J}\cup\{u_{g(i)}\}} C_{(\mathcal{J}\setminus\{k\})\cup\{u_{g(i)}\},(\mathcal{B}\setminus\{i\})\cup\{d_k\}}=0, \tag{57}$$

for any $i\in\mathcal{B}$ where $u_{g(i)}\notin\mathcal{J}$. We define $\mathcal{R}=\mathcal{J}\cup\{u_{g(i)}\}$. Since $u_{g(i)}\notin\mathcal{J}$ and $|\mathcal{J}|=t+1$, we have $|\mathcal{R}|=t+2$. Any $C_{\mathcal{T},\mathcal{H}}$ in (57), should satisfy $\mathcal{T}\subseteq\mathcal{R}$ and $|\mathcal{R}\setminus\mathcal{T}|=1$. The desired file of the user in $\mathcal{R}\setminus\mathcal{T}$, is in $\mathcal{H}$. In addition, if $C_{\mathcal{T}_1,\mathcal{H}_1}$ and $C_{\mathcal{T}_2,\mathcal{H}_2}$ are in (57), we can see that $\mathcal{T}_1\neq\mathcal{T}_2$.

We focus one sub-block $W_{\mathcal{S},\mathcal{V}}$ in (57) and assume that $C_{\mathcal{T},\mathcal{H}}$ contains $W_{\mathcal{S},\mathcal{V}}$. Hence, we have $\mathcal{S}\subseteq\mathcal{N}_{\mathbf{d}}(\mathcal{T})\cup\mathcal{H}$, $\mathcal{V}\subseteq\mathcal{T}$, $|\mathcal{T}\setminus\mathcal{V}|=1$, and the user in $\mathcal{T}\setminus\mathcal{V}$ (assumed to be user k') desires the sub-block $W_{\mathcal{S},\mathcal{V}}$. In addition, since $k'\in\mathcal{T}\subseteq\mathcal{R}$ and $|\mathcal{R}\setminus\mathcal{T}|=1$, assuming $k_1\in\mathcal{R}\setminus\mathcal{T}$, we have $d_{k_1}\in\mathcal{H}$ and thus $W_{\mathcal{S},\mathcal{V}}$ is also desired by user k_1. Hence, it can be seen that $C_{\mathcal{V}\cup\{k_1\},\mathcal{H}\setminus\{d_{k_1}\}\cup\{d_{k'}\}}$ is also in (57), and $W_{\mathcal{S},\mathcal{V}}$ desired by user k_1 is in $C_{\mathcal{V}\cup\{k_1\},\mathcal{H}\setminus\{d_{k_1}\}\cup\{d_{k'}\}}$. Except $C_{\mathcal{T},\mathcal{H}}$ and $C_{\mathcal{V}\cup\{k_1\},\mathcal{H}\setminus\{d_{k_1}\}\cup\{d_{k'}\}}$, there does not exist any other $C_{\mathcal{T}_1,\mathcal{H}_1}$ in (57) containing $W_{\mathcal{S},\mathcal{V}}$; this is because except $\mathcal{T}$ and $\mathcal{V}\cup\{k_1\}$, there does not exist any other $\mathcal{T}_1\subseteq\mathcal{R}$ where $|\mathcal{T}_1|=|\mathcal{R}|-1$ and $\mathcal{V}\subseteq\mathcal{T}_1$ (noticing that $\mathcal{V}\subseteq\mathcal{R}$ and $|\mathcal{V}|=|\mathcal{R}|-2$).

In conclusion, each sub-block in (57) appears twice in (57), and thus we proved (57).

B. Proof of (34)

Focus on one $\mathcal{J}\subseteq[\mathsf{K}]$ where $|\mathcal{J}|=t+1$, and one $\mathcal{B}\subseteq[\mathsf{N}]$ where $|\mathcal{B}|=\mathsf{r}-1$ and $\mathcal{N}_{\mathbf{d}}(\mathcal{J})\cap\mathcal{B}\neq\emptyset$. To prove (34), it is equivalent to prove that, for any $i_1\in\mathcal{N}_{\mathbf{d}}(\mathcal{J})\cap\mathcal{B}$,

$$\bigoplus_{i\in(\mathcal{N}_{\mathbf{d}}(\mathcal{J})\setminus\mathcal{B})\cup\{i_1\}} C_{\mathcal{J},(\mathcal{B}\setminus\{i_1\})\cup\{i\}}=0. \tag{58}$$

Assume that $C_{\mathcal{J},\mathcal{H}}$ appears on the LHS of (58). Thus we have $(\mathcal{B}\setminus\{i_1\})\subseteq\mathcal{H}$ and $|\mathcal{H}\setminus(\mathcal{B}\setminus\{i_1\})|=1$. In addition, the file in $\mathcal{H}\setminus(\mathcal{B}\setminus\{i_1\})$, is also in $(\mathcal{N}_{\mathbf{d}}(\mathcal{J})\setminus\mathcal{B})\cup\{i_1\}$.

We focus one sub-block $W_{\mathcal{S},\mathcal{V}}$, which is in $C_{\mathcal{J},\mathcal{H}}$. Thus $\mathcal{H}\subseteq\mathcal{S}$ and $|\mathcal{S}\setminus\mathcal{H}|=1$ (we assume the file in $\mathcal{S}\setminus\mathcal{H}$ is i'). In addition, we have $(\mathcal{B}\setminus\{i_1\})\subseteq\mathcal{H}$ and $|\mathcal{H}\setminus(\mathcal{B}\setminus\{i_1\})|=1$ (we assume the file in $\mathcal{H}\setminus(\mathcal{B}\setminus\{i_1\})$ is i''). As described before, i'' is $(\mathcal{N}_{\mathbf{d}}(\mathcal{J})\setminus\mathcal{B})\cup\{i_1\}$ and thus file $F_{i''}$ is demanded by some user in $\mathcal{J}$ (recall that $i_1\in\mathcal{N}_{\mathbf{d}}(\mathcal{J})\cap\mathcal{B}$). Hence, it can be seen that $W_{\mathcal{S},\mathcal{V}}$ is also in $C_{\mathcal{J},\mathcal{H}\setminus\{i''\}\cup\{i'\}}$. Except $C_{\mathcal{J},\mathcal{H}}$ and $C_{\mathcal{J},\mathcal{H}\setminus\{i''\}\cup\{i'\}}$, there does not exist any other $C_{\mathcal{J},\mathcal{H}_1}$ in (58) containing $W_{\mathcal{S},\mathcal{V}}$; this is because except $\mathcal{H}$ and $\mathcal{H}\setminus\{i''\}\cup\{i'\}$ there does not exist any other $\mathcal{H}_1\subseteq\mathcal{S}$ where $(\mathcal{B}\setminus\{i_1\})\subseteq\mathcal{H}_1$ and $|\mathcal{H}_1|=|\mathcal{S}|-1$ (noticing that $(\mathcal{B}\setminus\{i_1\})\subseteq\mathcal{S}$ and $|\mathcal{B}\setminus\{i_1\}|=|\mathcal{S}|-2$).

In conclusion, each sub-block in (58) appears twice in (58), and thus we proved (58).

Appendix D
Proof of Lemma 4

We use the induction method to prove Lemma 4.

$j=1$. We will prove that each user can reconstruct $C_{\mathcal{J}\cup\{u_1\},\mathcal{B}\cup\{d_{u_1}\}}$ where $\mathcal{J}\subseteq[\mathsf{K}]\setminus\{u_1\}$, $|\mathcal{J}|=t$, $\mathcal{B}\subseteq[\mathsf{N}]\setminus\{d_{u_1}\}$, $|\mathcal{B}|=\mathsf{r}-2$, and $\mathcal{N}_{\mathbf{d}}(\mathcal{J}\cap\mathcal{L}(\mathbf{d}))\setminus\mathcal{B}\neq\emptyset$.

By (34) in Lemma 3, we have

$$C_{\mathcal{J}\cup\{u_1\},\mathcal{B}\cup\{d_{u_1}\}}=\bigoplus_{i_2\in\mathcal{N}_{\mathbf{d}}(\mathcal{J})\setminus(\mathcal{B}\cup\{d_{u_1}\})} C_{\mathcal{J}\cup\{u_1\},\mathcal{B}\cup\{i_2\}}, \tag{59}$$

where each $C_{\mathcal{J}\cup\{u_1\},\mathcal{B}\cup\{i_2\}}$ is transmitted in Step $j=1$ of the first sub-phase. Hence, each user can reconstruct $C_{\mathcal{J}\cup\{u_1\},\mathcal{B}\cup\{d_{u_1}\}}$.

Authorized licensed use limited to: University of Illinois at Chicago Library. Downloaded on November 08,2023 at 23:24:46 UTC from IEEE Xplore. Restrictions apply.

$j \in [2 : \min\{N_{\mathrm{e}}(\mathbf{d}), \mathsf{N}-\mathsf{r}+1, \mathsf{K}-t\}]$. Assume that for each $j' \in [j-1]$ and $i' \in \{d_{u_1}, \ldots, d_{u_{j'}}\}$, each user has reconstructed $C_{\mathcal{J}'\cup\{u_{j'}\},\mathcal{B}'\cup\{i'\}}$ where $\mathcal{J}' \subseteq [\mathsf{K}] \setminus \{u_1, \ldots, u_{j'}\}$, $|\mathcal{J}'| = t$, $\mathcal{B}' \subseteq [\mathsf{N}] \setminus \{d_{u_1}, \ldots, d_{u_{j'}}\}$, $|\mathcal{B}'| = \mathsf{r}-2$, and $\mathcal{N}_{\mathbf{d}}(\mathcal{J}' \cap \mathcal{L}(\mathbf{d})) \setminus \mathcal{B}' \neq \emptyset$.

Now for $i \in \{d_{u_1}, \ldots, d_{u_j}\}$, we want to prove that each user can reconstruct $C_{\mathcal{J}\cup\{u_j\},\mathcal{B}\cup\{i\}}$ where $\mathcal{J} \subseteq [\mathsf{K}] \setminus \{u_1, \ldots, u_j\}$, $|\mathcal{J}| = t$, $\mathcal{B} \subseteq [\mathsf{N}] \setminus \{d_{u_1}, \ldots, d_{u_j}\}$, $|\mathcal{B}| = \mathsf{r}-2$, and $\mathcal{N}_{\mathbf{d}}(\mathcal{J} \cap \mathcal{L}(\mathbf{d})) \setminus \mathcal{B} \neq \emptyset$.

We first consider the case where $i \in \{d_{u_1}, \ldots, d_{u_{j-1}}\}$. By (33) in Lemma 3, we have

$$C_{\mathcal{J}\cup\{u_j\},\mathcal{B}\cup\{i\}} = \underset{k\in\mathcal{J}\cup\{u_j\}}{\oplus} C_{\mathcal{J}\cup\{u_j,u_{g(i)}\}\setminus\{k\},\mathcal{B}\cup\{d_k\}}. \quad (60)$$

For each $k \in \mathcal{J} \cup \{u_j\}$,

- if $d_k \in \{d_{u_1}, \ldots, d_{u_{j-1}}\}$, each user can reconstruct $C_{\mathcal{J}\cup\{u_j,u_{g(i)}\}\setminus\{k\},\mathcal{B}\cup\{d_k\}}$ by the induction assumption (by letting $j' = u_{g(i)}$, $i' = d_k$, $\mathcal{J}' = \mathcal{J}\cup\{u_j\}\setminus\{k\}$, and $\mathcal{B}' = \mathcal{B}$);
- if $d_k \notin \{d_{u_1}, \ldots, d_{u_{j-1}}\}$ and $d_k \notin \mathcal{B}$, $C_{\mathcal{J}\cup\{u_j,u_{g(i)}\}\setminus\{k\},\mathcal{B}\cup\{d_k\}}$ is transmitted in Step $g(i)$ of the first sub-phase;
- if $d_k \notin \{d_{u_1}, \ldots, d_{u_{j-1}}\}$ and $d_k \in \mathcal{B}$, since $\mathcal{N}_{\mathbf{d}}(\mathcal{J} \cap \mathcal{L}(\mathbf{d})) \setminus \mathcal{B} \neq \emptyset$, we can see that there exists one leader in $\mathcal{J}\setminus\{k\}$ whose demanded file is not in $\mathcal{B}$. Thus in this case, $C_{\mathcal{J}\cup\{u_j,u_{g(i)}\}\setminus\{k\},\mathcal{B}\cup\{d_k\}} = C_{\mathcal{J}\cup\{u_j,u_{g(i)}\}\setminus\{k\},\mathcal{B}}$ is transmitted in Step $g(i)$ of the second sub-phase.

Hence, each user can recover $C_{\mathcal{J}\cup\{u_j\},\mathcal{B}\cup\{i\}}$.

We then focus on the case $i = d_{u_j}$, and consider $C_{\mathcal{J}\cup\{u_j\},\mathcal{B}\cup\{d_{u_j}\}}$. By (34) in Lemma 3, we have

$$C_{\mathcal{J}\cup\{u_j\},\mathcal{B}\cup\{d_{u_j}\}} = \underset{i_2\in\mathcal{N}_{\mathbf{d}}(\mathcal{J})\setminus(\mathcal{B}\cup\{d_{u_j}\})}{\oplus} C_{\mathcal{J}\cup\{u_j\},\mathcal{B}\cup\{i_2\}}. \quad (61)$$

On the RHS of (61), if $i_2 \in \{d_{u_1}, \ldots, d_{u_{j-1}}\}$, it has been proved in (60) that $C_{\mathcal{J}\cup\{u_j\},\mathcal{B}\cup\{i_2\}}$ can be reconstructed by each user; otherwise, $C_{\mathcal{J}\cup\{u_j\},\mathcal{B}\cup\{i_2\}}$ is transmitted in Step j of the first sub-phase. Hence, each user can recover $C_{\mathcal{J}\cup\{u_j\},\mathcal{B}\cup\{d_{u_j}\}}$.

Remark 5: Notice that to prove Lemma 4, the condition $\mathcal{N}_{\mathbf{d}}(\mathcal{J} \cap \mathcal{L}(\mathbf{d})) \setminus \mathcal{B} \neq \emptyset$ and the transmission in the second sub-phase are only used when there exists some user $k \in \mathcal{J}\cup\{u_j\}$ whose demanded file is in $\mathcal{B}$ (i.e., $d_k \in \mathcal{B}$ in (60)).

Hence, for each $j \in [\min\{N_{\mathrm{e}}(\mathbf{d}), \mathsf{N}-\mathsf{r}+1, \mathsf{K}-t\}]$ and each $i \in \{d_{u_1}, \ldots, d_{u_j}\}$, by using the transmission in the first sub-phase, each user can recover $C_{\mathcal{J}\cup\{u_j\},\mathcal{B}\cup\{i\}}$, where $\mathcal{J} \subseteq [\mathsf{K}] \setminus \{u_1, \ldots, u_j\}$, $|\mathcal{J}| = t$, $\mathcal{B} \subseteq ([\mathsf{N}] \setminus \{d_{u_1}, \ldots, d_{u_j}\})$, $|\mathcal{B}| = \mathsf{r}-2$, and $\mathcal{N}_{\mathbf{d}}(\mathcal{J}) \cap \mathcal{B} = \emptyset$. □

APPENDIX E
PROOF OF DECODABILITY OF THE GENERAL SCHEME IN SECTION V-C

Now we are ready to prove the decodability of each non-leader k. In other words, we want to prove that it can decode $W_{\mathcal{S},\mathcal{V}}$ where $d_k \in \mathcal{S}$, $\{d_{u_1}, \ldots, d_{u_{g(d_k)-1}}\} \cap \mathcal{S} = \emptyset$ and $\{k, u_1, \ldots, u_{g(d_k)}\} \cap \mathcal{V} = \emptyset$ (in Lemma 1 we showed that the other desired sub-blocks could be decoded by user k from the transmission of the first sub-phase). We consider two cases, $|\mathcal{S} \cap \mathcal{N}_{\mathbf{d}}([\mathsf{K}])| > 1$ and $|\mathcal{S} \cap \mathcal{N}_{\mathbf{d}}([\mathsf{K}])| = 1$.

A. $|\mathcal{S} \cap \mathcal{N}_{\mathbf{d}}([\mathsf{K}])| > 1$

Among all desired sub-blocks in this case, we use the induction method to prove for each $j \in [g(d_k)+1 : \min\{N_{\mathrm{e}}(\mathbf{d}), \mathsf{N}-\mathsf{r}+2, \mathsf{K}-t+1\}]$, user k can recover its desired sub-blocks $W_{\mathcal{S},\mathcal{V}}$ (i.e., $d_k \in \mathcal{S}$) where $d_{u_j} \in \mathcal{S}$ or $u_j \in \mathcal{V}$.

Induction on $j = g(d_k)+1$. We consider three cases:

- $u_j \in \mathcal{V}$ and $d_{u_j} \notin \mathcal{S}$. In $C_{\mathcal{V}\cup\{k\},\mathcal{S}\setminus\{d_k\}}$ transmitted in Step j of the first sub-phase, user k caches all sub-blocks except $W_{\mathcal{S},\mathcal{V}}$ and thus it can recover $W_{\mathcal{S},\mathcal{V}}$ by directly reading off.
- $u_j \in \mathcal{V}$ and $d_{u_j} \in \mathcal{S}$. Since $u_j \in \mathcal{V}$, by Lemma 4 it can be seen that user k can reconstruct $C_{\mathcal{V}\cup\{u_{g(d_k)}\},\mathcal{S}\setminus\{d_{u_j}\}}$ at the end of Step $g(d_k)$ of sub-phase 2.[12]
  In $C_{\mathcal{V}\cup\{u_{g(d_k)}\},\mathcal{S}\setminus\{d_{u_j}\}}$, all sub-blocks are desired by user k. In addition, all sub-blocks desired by user k which are cached by user $u_{g(d_k)}$, can be recovered by user k by Lemma 1.Item 2.
  The sub-blocks in $C_{\mathcal{V}\cup\{u_{g(d_k)}\},\mathcal{S}\setminus\{d_{u_j}\}}$ which are not cached by user $u_{g(d_k)}$, are all cached by user u_j (because $u_j \in \mathcal{V}$). For each $i \in \mathcal{N}_{\mathbf{d}}(\mathcal{V}) \setminus (\mathcal{S}\setminus\{d_{u_j}\})$, the sub-block $W_{\mathcal{S}\setminus\{d_{u_j}\}\cup\{i\},\mathcal{V}}$ is in $C_{\mathcal{V}\cup\{u_{g(d_k)}\},\mathcal{S}\setminus\{d_{u_j}\}}$ which is desired (and not cached) by user $u_{g(d_k)}$.
  If $i \neq d_{u_j}$, since $d_{u_j} \notin (\mathcal{S}\setminus\{d_{u_j}\}\cup\{i\})$ and $u_j \in \mathcal{V}$, we proved in the first case that $W_{\mathcal{S}\setminus\{d_{u_j}\}\cup\{i\},\mathcal{V}}$ can be recovered by user k; otherwise, the sub-block $W_{\mathcal{S}\setminus\{d_{u_j}\}\cup\{i\},\mathcal{V}}$ is $W_{\mathcal{S},\mathcal{V}}$. Hence, in $C_{\mathcal{V}\cup\{u_{g(d_k)}\},\mathcal{S}\setminus\{d_{u_j}\}}$, only sub-block $W_{\mathcal{S},\mathcal{V}}$ is not recovered by user k, such that user k can recover $W_{\mathcal{S},\mathcal{V}}$.
- $u_j \notin \mathcal{V}$ and $d_{u_j} \in \mathcal{S}$. We first prove that user k can reconstruct $C_{\mathcal{V}\cup\{u_j\},\mathcal{S}\setminus\{d_{u_j}\}}$. From (33) in Lemma 3, we have

$$C_{\mathcal{V}\cup\{u_j\},\mathcal{S}\setminus\{d_{u_j}\}} = \underset{k_1\in(\mathcal{V}\cup\{u_j\})}{\oplus} C_{\mathcal{V}\cup\{u_j,u_{g(d_k)}\}\setminus\{k_1\},(\mathcal{S}\setminus\{d_{u_j},d_k\})\cup\{d_{k_1}\}}. \quad (62)$$

  For each $k_1 \in (\mathcal{V}\cup\{u_j\})$ in (62),
  - sub-case 1: if $k_1 = u_j$, we have

$$C_{\mathcal{V}\cup\{u_j,u_{g(d_k)}\}\setminus\{k_1\},(\mathcal{S}\setminus\{d_{u_j},d_k\})\cup\{d_{k_1}\}} = C_{\mathcal{V}\cup\{u_{g(d_k)}\},\mathcal{S}\setminus\{d_k\}},$$

    which is transmitted in Step $g(d_k)$ of the first sub-phase;
  - sub-case 2: if $k_1 \neq u_j$ and $d_{k_1} \notin \{d_{u_1}, \ldots, d_{u_{g(d_k)}}\}$, it can be seen that $C_{\mathcal{V}\cup\{u_j,u_{g(d_k)}\}\setminus\{k_1\},(\mathcal{S}\setminus\{d_{u_j},d_k\})\cup\{d_{k_1}\}}$ is transmitted either in Step $g(d_k)$ of the first sub-phase (if $|(\mathcal{S}\setminus\{d_{u_j},d_k\})\cup\{d_{k_1}\}| = \mathsf{r}-1$) or Step $g(d_k)$ of the second sub-phase (if $|(\mathcal{S}\setminus\{d_{u_j},d_k\})\cup\{d_{k_1}\}| = \mathsf{r}-2$ and $(\mathcal{V}\setminus\{k_1\}) \cap \mathcal{N}_{\mathbf{d}}([\mathsf{K}]) \neq \emptyset$) or Step $g(d_k)$ in

[12] From the proof of Lemma 4 in Appendix D, to reconstruct $C_{\mathcal{V}\cup\{u_{g(d_k)}\},\mathcal{S}\setminus\{d_{u_j}\}}$, user k only needs to use sub-phase 1 and Step $g(d_k)$ of sub-phase 2. This is because $d_k \in \mathcal{S}\setminus\{d_{u_j}\}$ and $\{d_{u_1}, \ldots, d_{u_{g(d_k)-1}}\} \cap \mathcal{S} = \emptyset$; thus in (60), we have $i = d_k$.

Authorized licensed use limited to: University of Illinois at Chicago Library. Downloaded on November 08,2023 at 23:24:46 UTC from IEEE Xplore. Restrictions apply.

Lemma 2 (if $|(\mathcal{S}\setminus\{d_{u_j},d_k\})\cup\{d_{k_1}\}|=\mathsf{r}-2$ and $(\mathcal{V}\setminus\{k_1\})\cap\mathcal{N}_{\mathbf{d}}([\mathsf{K}])=\emptyset$);

- sub-case 3: if $k_1\neq u_j$ and $d_{k_1}\in\{d_{u_1},\ldots,d_{u_{g(d_k)}}\}$, by Lemma 4, $C_{\mathcal{V}\cup\{u_j,u_{g(d_k)}\}\setminus\{k_1\},(\mathcal{S}\setminus\{d_{u_j},d_k\})\cup\{d_{k_1}\}}$ can be reconstructed by user k at the end of Step $g(d_k)$ of sub-phase 2.[13]

Hence, user k can recover each message on the RHS of (62) and thus it can reconstruct $C_{\mathcal{V}\cup\{u_j\},\mathcal{S}\setminus\{d_{u_j}\}}$.

In $C_{\mathcal{V}\cup\{u_j\},\mathcal{S}\setminus\{d_{u_j}\}}$, all sub-blocks are desired by user k. For each $k_2\in(\mathcal{V}\cup\{u_j\})$, if $k_2\neq u_j$, the desired sub-blocks in $C_{\mathcal{V}\cup\{u_j\},\mathcal{S}\setminus\{d_{u_j}\}}$ by user k_2 are stored by user u_j, which can be recovered by user k from the transmission of the first sub-phase (as we proved above for the case $u_j\in\mathcal{V}$ and $d_{u_j}\notin\mathcal{S}$). If $k_2=u_j$, the desired sub-block by user k_2 is $W_{\mathcal{S},\mathcal{V}}$. Hence, user k can recover $W_{\mathcal{S},\mathcal{V}}$.

Induction on $j\in[g(d_k)+2:\min\{N_{\mathrm{e}}(\mathbf{d}),\mathsf{N}-\mathsf{r}+2,\mathsf{K}-t+1\}]$. If there exists $j'\in[g(d_k)+1:j-1]$, where $u_{j'}\in\mathcal{V}$ or $d_{u_{j'}}\in\mathcal{S}$, by the induction assumption, user k can recover $W_{\mathcal{S},\mathcal{V}}$; otherwise, we can use the similar proof by dividing into three cases and using the induction assumption, to prove user k can recover $W_{\mathcal{S},\mathcal{V}}$ (for the sake of simplicity, we do not repeat).

Remark 6: If there exists one leader in $\mathcal{V}$ (assumed to be k') where $d_{k'}\notin\mathcal{S}$, we can prove that user k can recover $W_{\mathcal{S},\mathcal{V}}$ without using Lemma 2.

More precisely, we focus on the case $u_j\in\mathcal{V}$ and $d_{u_j}\in\mathcal{S}$, where Lemma 2 may be needed. In (62), for each $k_1\in\mathcal{V}$ where $d_{k_1}\notin\{d_{u_1},\ldots,d_{u_{g(d_k)}}\}$, if $k_1\neq k'$, it can be seen that $(\mathcal{V}\setminus\{k_1\})\cap\mathcal{N}_{\mathbf{d}}([\mathsf{K}])\neq\emptyset$ and thus Lemma 2 is not needed; otherwise, we have $k_1=k'$ and $|(\mathcal{S}\setminus\{d_{u_j},d_k\})\cup\{d_{k'}\}|=\mathsf{r}-1$ such that Lemma 2 is not needed neither. □

B. $|\mathcal{S}\cap\mathcal{N}_{\mathbf{d}}([\mathsf{K}])|=1$

We can gather all blocks $W_{\mathcal{S}'}$ where $\mathcal{S}'\cap([\mathsf{N}]\setminus\mathcal{N}_{\mathbf{d}}([\mathsf{K}]))=\mathcal{S}\cap([\mathsf{N}]\setminus\mathcal{N}_{\mathbf{d}}([\mathsf{K}]))$. The transmission for these blocks is equivalent to the MAN caching problem in [2] and thus from the transmission of the first sub-phase on these blocks which is equivalent to the optimal caching scheme in [4], each non-leader can recover $W_{\mathcal{S},\mathcal{V}}$.

C. Proof of Observations

Proof of Observation 1: Recall that in Step $j\in[\min\{N_{\mathrm{e}}(\mathbf{d}),\mathsf{N}-\mathsf{r}+1,\mathsf{K}-t\}]$ of sub-phase 2, we transmit $C_{\mathcal{J},\mathcal{B}}$ where $q\in[j+1:\min\{\mathsf{N}-\mathsf{r}+2,\mathsf{K}-t+1,N_{\mathrm{e}}(\mathbf{d})\}]$, $\mathcal{J}\subseteq([\mathsf{K}]\setminus\{u_1,\ldots,u_{q-1}\}\cup\{u_j\})$, $|\mathcal{J}|=t+1$, $\{u_j,u_q\}\subseteq\mathcal{J}$, $\mathcal{J}\cap\{u_{q+1},\ldots,u_{N_{\mathrm{e}}(\mathbf{d})}\}\neq\emptyset$, $\mathcal{B}\subseteq[\mathsf{N}]\setminus\{d_{u_1},\ldots,d_{u_q}\}$, $|\mathcal{B}|=\mathsf{r}-2$, and $\mathcal{B}\cap\mathcal{N}_{\mathbf{d}}([\mathsf{K}])\neq\emptyset$.

When $\mathsf{r}=2$, the transmission of the second sub-phase does not exist because $|\mathcal{B}|=\mathsf{r}-2=0$ and $\mathcal{B}\cap\mathcal{N}_{\mathbf{d}}([\mathsf{K}])\neq\emptyset$ cannot hold simultaneously.

[13]From the proof of Lemma 4 in Appendix D, to reconstruct $C_{\mathcal{V}\cup\{u_j,u_{g(d_k)}\}\setminus\{k_1\},(\mathcal{S}\setminus\{d_{u_j},d_k\})\cup\{d_{k_1}\}}$, user k only needs to use sub-phase 1 and Step $g(d_{k_1})$ of sub-phase 2. This is because $d_{k_1}\in\{d_{u_1},\ldots,d_{u_{g(d_k)}}\}$ and $\{d_{u_1},\ldots,d_{u_{g(d_k)-1}}\}\cap\mathcal{S}=\emptyset$; thus in (60), we have $i=d_{k_1}$.

When $t=1$, the transmission of the second sub-phase does not exist because when $|\mathcal{J}|=t+1=2$, $\{u_j,u_q\}\subseteq\mathcal{J}$ and $\mathcal{J}\cap\{u_{q+1},\ldots,u_{N_{\mathrm{e}}(\mathbf{d})}\}\neq\emptyset$, cannot hold simultaneously.

Proof of Observation 2: We want to prove that for a non-leader k, to decode $W_{\mathcal{S},\mathcal{V}}$ where $d_k\in\mathcal{S}$, $\{d_{u_1},\ldots,d_{u_{g(d_k)-1}}\}\cap\mathcal{S}=\emptyset$ and $\{k,u_1,\ldots,u_{g(d_k)}\}\cap\mathcal{V}=\emptyset$, if there is no user in $\mathcal{V}$ whose demanded file is in $\{d_{u_1},\ldots,d_{u_{g(d_k)-1}}\}$, user k only needs to use the transmission of the first sub-phase, Step $g(d_k)$ of the second sub-phase and Step $g(d_k)$ in Lemma 2.

Besides the transmission of the first sub-phase, Step $g(d_k)$ of the second sub-phase and Step $g(d_k)$ in Lemma 2, other steps of the second sub-phase may be needed only when we use Lemma 4 to show that non-leader k can reconstruct (sub-case 3 in (62))

$$C_{\mathcal{V}\cup\{u_j,u_{g(d_k)}\}\setminus\{k_1\},(\mathcal{S}\setminus\{d_{u_j},d_k\})\cup\{d_{k_1}\}}=C_{\mathcal{V}\cup\{u_j,u_{g(d_k)}\}\setminus\{k_1\},(\mathcal{S}\setminus\{d_{u_j}\}},$$

where $d_{k_1}\in\{d_{u_1},\ldots,d_{u_{g(d_k)-1}}\}$, as explained in Footnote 13. Hence, if there is no user in $\mathcal{V}$ whose demanded file is in $\{d_{u_1},\ldots,d_{u_{g(d_k)-1}}\}$, non-leader k does not need to use the transmission of other steps of the second sub-phase; thus we proved Observation 2.

Proof of Observation 3: We want to prove that, for a non-leader k, to decode $W_{\mathcal{S},\mathcal{V}}$ where $d_k\in\mathcal{S}$, $\{d_{u_1},\ldots,d_{u_{g(d_k)-1}}\}\cap\mathcal{S}=\emptyset$, $\{k,u_1,\ldots,u_{g(d_k)}\}\cap\mathcal{V}=\emptyset$, and $(\cup_{k'\in\mathcal{V}}\{d_{k'}\})\cap(\mathcal{S}\setminus\{d_k\})=\emptyset$, user k only needs the transmission of the first sub-phase.

If $|\mathcal{S}\cap\mathcal{N}_{\mathbf{d}}([\mathsf{K}])|=1$, it has been proved that only the first sub-phase is needed. Hence, in the following we consider $|\mathcal{S}\cap\mathcal{N}_{\mathbf{d}}([\mathsf{K}])|>1$. We focus on the induction Step $j\in[g(d_k)+1:\min\{N_{\mathrm{e}}(\mathbf{d}),\mathsf{N}-\mathsf{r}+2,\mathsf{K}-t+1\}]$ in the decodability proof in Appendix E-A, and consider the following cases:

- if $u_j\in\mathcal{V}$ and $d_{u_j}\notin\mathcal{S}$, from the proof in Appendix E-A, the first sub-phase is only needed;
- if $u_j\in\mathcal{V}$ and $d_{u_j}\in\mathcal{S}$, user k needs to reconstruct $C_{\mathcal{V}\cup\{u_{g(d_k)}\},\mathcal{S}\setminus\{d_{u_j}\}}$. Since $(\cup_{k'\in\mathcal{V}}\{d_{k'}\})\cap(\mathcal{S}\setminus\{d_k\})=\emptyset$, from Remark 5 we can see that $C_{\mathcal{V}\cup\{u_{g(d_k)}\},\mathcal{S}\setminus\{d_{u_j}\}}$ can be reconstructed by user k from the transmission of the first sub-phase;
- finally we focus on $u_j\notin\mathcal{V}$ and $d_{u_j}\in\mathcal{S}$. In this case, user k needs to recover the LHS of (62). On the RHS of (62), for each $k_1\in(\mathcal{V}\cup\{u_j\})$,
  - if $k_1=u_j$, the first sub-phase is only needed;
  - if $k_1\neq u_j$ and $d_{k_1}\notin\{d_{u_1},\ldots,d_{u_{g(d_k)}}\}$, since $(\cup_{k'\in\mathcal{V}}\{d_{k'}\})\cap(\mathcal{S}\setminus\{d_k\})=\emptyset$, we have $|(\mathcal{S}\setminus\{d_{u_j},d_k\})\cup\{d_{k_1}\}|=\mathsf{r}-1$ and thus we only need the first sub-phase;
  - if $k_1\neq u_j$ and $d_{k_1}\in\{d_{u_1},\ldots,d_{u_{g(d_k)}}\}$, user k should reconstruct $C_{\mathcal{V}\cup\{u_j,u_{g(d_k)}\}\setminus\{k_1\},(\mathcal{S}\setminus\{d_{u_j},d_k\})\cup\{d_{k_1}\}}$. Since $(\cup_{k'\in\mathcal{V}}\{d_{k'}\})\cap(\mathcal{S}\setminus\{d_k\})=\emptyset$, from Remark 5 we can see that user k can reconstruct $C_{\mathcal{V}\cup\{u_j,u_{g(d_k)}\}\setminus\{k_1\},(\mathcal{S}\setminus\{d_{u_j},d_k\})\cup\{d_{k_1}\}}$ from the first sub-phase.

Hence, we proved Observation 3.

Authorized licensed use limited to: University of Illinois at Chicago Library. Downloaded on November 08,2023 at 23:24:46 UTC from IEEE Xplore. Restrictions apply.

APPENDIX F
PROOF OF THE DECODABILITY FOR $\mathsf{r} = \mathsf{N} - 1$ OR $t = 2$

We now consider $\mathsf{r} = \mathsf{N} - 1$ or $t = 2$ and prove that each non-leader k can recover $W_{\mathcal{S},\mathcal{V}}$ where $d_k \in \mathcal{S}$, $\{d_{u_1}, \ldots, d_{u_{g(d_k)-1}}\} \cap \mathcal{S} = \emptyset$ and $\{k, u_1, \ldots, u_{g(d_k)}\} \cap \mathcal{V} = \emptyset$, from the first sub-phase. If $\mathcal{N}_{\mathbf{d}}(\mathcal{V}) \cap (\mathcal{S} \setminus \{d_k\}) = \emptyset$, by Observation 3, user k can recover $W_{\mathcal{S},\mathcal{V}}$ from the first sub-phase. Hence, in the following, we focus on $\mathcal{N}_{\mathbf{d}}(\mathcal{V}) \cap (\mathcal{S} \setminus \{d_k\}) \neq \emptyset$. We consider two cases, $\mathcal{N}_{\mathbf{d}}(\mathcal{V}) \cap \{d_{u_1}, \ldots, d_{u_{g(d_k)-1}}\} = \emptyset$ and $\mathcal{N}_{\mathbf{d}}(\mathcal{V}) \cap \{d_{u_1}, \ldots, d_{u_{g(d_k)-1}}\} \neq \emptyset$, respectively.

A. $\mathcal{N}_{\mathbf{d}}(\mathcal{V}) \cap \{d_{u_1}, \ldots, d_{u_{g(d_k)-1}}\} = \emptyset$

By Observation 2, if user k obtain the multicast messages in the first sub-phase, in Step $g(d_k)$ of the second sub-phase, and in Step $g(d_k)$ in Lemma 2, it can recover $W_{\mathcal{S},\mathcal{V}}$. In the following, we will prove that user k can reconstruct the multicast messages in Step $g(d_k)$ of the second sub-phase and in Step $g(d_k)$ in Lemma 2 by using the transmission of the first sub-phase. In other words, we will prove that for each integer $q \in [g(d_k)+1 : \min\{\mathsf{N}-\mathsf{r}+2, \mathsf{K}-t+1, N_e(\mathbf{d})\}]$, user k can reconstruct $C_{\mathcal{J},\mathcal{B}}$ from the first sub-phase, where $\mathcal{J} \subseteq ([\mathsf{K}] \setminus \{u_1, \ldots, u_{q-1}\} \cup \{u_{g(d_k)}\})$, $|\mathcal{J}| = t+1$, $\{u_{g(d_k)}, u_q\} \subseteq \mathcal{J}$, $\mathcal{B} \subseteq [\mathsf{N}] \setminus \{d_{u_1}, \ldots, d_{u_q}\}$, $|\mathcal{B}| = \mathsf{r}-2$, and $\mathcal{B} \cap \mathcal{N}_{\mathbf{d}}([\mathsf{K}]) \neq \emptyset$.

If there is no user in $\mathcal{J} \setminus \{u_{g(d_k)}, u_q\}$ whose demand is in $[\mathsf{N}] \setminus (\{d_k, d_{u_q}\} \cup \mathcal{B})$, it can be seen that all sub-blocks in $C_{\mathcal{J},\mathcal{B}}$ are from $W_{\mathcal{B}\cup\{d_k,d_{u_q}\}}$. Hence, we have $C_{\mathcal{J},\mathcal{B}} = C_{\mathcal{J},\mathcal{B}\cup\{d_{u_q}\}}$, which is transmitted in Step $g(d_k)$ of the first sub-phase. Hence, in the following, we consider that there exists some user in $\mathcal{J} \setminus \{u_{g(d_k)}, u_q\}$ whose demand is in $[\mathsf{N}] \setminus (\{d_k, d_{u_q}\} \cup \mathcal{B})$.

For the case $t = 2$, we have $|\mathcal{J} \setminus \{u_{g(d_k)}, u_q\}| = 1$; for the case $\mathsf{r} = \mathsf{N}-1$, we have $|\mathcal{B}| = \mathsf{r}-2 = \mathsf{N}-3$ and thus $|[\mathsf{N}] \setminus (\mathcal{B} \cup \{d_k, d_{u_q}\})| = 1$. Hence, when $\mathsf{r} = \mathsf{N}-1$ or $t = 2$, there is only one file in $[\mathsf{N}] \setminus (\{d_k, d_{u_q}\} \cup \mathcal{B})$, which is demanded by some user in $\mathcal{J} \setminus \{u_{g(d_k)}, u_q\}$. We assume that this file is i. It can be seen that all interferences in $C_{\mathcal{J},\mathcal{B}}$ to user k, are from one block $W_{\mathcal{B}\cup\{d_{u_q},i\}}$. The sum of the interferences in $C_{\mathcal{J},\mathcal{B}}$ to user k is

$$I = \bigoplus_{k_1 \in \mathcal{J}\setminus\{u_{g(d_k)}\}: d_{k_1} \neq d_k} W_{\mathcal{B}\cup\{d_{u_q},i\},\mathcal{J}\setminus\{k_1\}}. \tag{63}$$

In addition, we also have

$$C_{\mathcal{J},\mathcal{B}} = C_{\mathcal{J},\mathcal{B}\cup\{d_{u_q}\}} \oplus \bigoplus_{k_2 \in \mathcal{J}\setminus\{u_q\}: d_{k_2} \neq d_{u_q}} W_{\mathcal{B}\cup\{d_k,i\},\mathcal{J}\setminus\{k_2\}}. \tag{64}$$

We then consider the following cases:

- if $i \notin \{d_{u_1}, \ldots, d_{u_{q-1}}\}$, in (64), $C_{\mathcal{J},\mathcal{B}\cup\{d_{u_q}\}}$ is transmitted in Step $g(d_k)$ of the first sub-phase.
  - If $k_2 \neq u_{g(d_k)}$, the sub-block $W_{\mathcal{B}\cup\{d_k,i\},\mathcal{J}\setminus\{k_2\}}$ is desired by user k and cached by user $u_{g(d_k)}$. Thus by Lemma 1.Item 2, user k can recover this sub-block from the transmission of the first sub-phase;
  - if $k_2 = u_{g(d_k)}$, $W_{\mathcal{B}\cup\{d_k,i\},\mathcal{J}\setminus\{u_{g(d_k)}\}}$ can be recovered by user k from $C_{\mathcal{J}\cup\{k\}\setminus\{u_{g(d_k)}\},\mathcal{B}\cup\{i\}}$ transmitted in Step q of the first sub-phase, where in $C_{\mathcal{J}\cup\{k\}\setminus\{u_{g(d_k)}\},\mathcal{B}\cup\{i\}}$ user k caches all except $W_{\mathcal{B}\cup\{d_k,i\},\mathcal{J}\setminus\{u_{g(d_k)}\}}$ such that it can recover this sub-block.

  Hence, user k can reconstruct $C_{\mathcal{J},\mathcal{B}}$ from the transmission of the first sub-phase;
- if $i \in \{d_{u_1}, \ldots, d_{u_{g(d_k)-1}}\}$, by Lemma 1.Item 3, we can see that each sub-block $W_{\mathcal{B}\cup\{d_k,i\},\mathcal{J}\setminus\{k_2\}}$ in (64) is from $W_{\mathcal{B}\cup\{d_k,i\}}$, which can be recovered by user k from the first sub-phase. Hence, user k can reconstruct $C_{\mathcal{J},\mathcal{B}}$ from the transmission of the first sub-phase;
- if $i \in \{d_{g(d_k)+1}, \ldots, d_{u_{q-1}}\}$, for each user $k_3 \in \mathcal{J} \setminus \{u_{g(d_k)}\}$ where $d_{k_3} \neq d_k$, we focus on $C_{\mathcal{J}\cup\{u_{g(i)}\}\setminus\{k_3\},\mathcal{B}\cup\{d_{u_q}\}}$ which is transmitted in Step $g(d_k)$ of the first sub-phase. In $C_{\mathcal{J}\cup\{u_{g(i)}\}\setminus\{k_3\},\mathcal{B}\cup\{d_{u_q}\}}$, since we have $|\mathcal{J} \setminus \{u_{g(d_k)}, u_q\}| = 1$ (for the case $t = 2$) or $|\mathcal{B}| = \mathsf{N}-3$ (for the case $\mathsf{r} = \mathsf{N}-1$), it can be seen that all sub-blocks are from either $W_{\mathcal{B}\cup\{d_{u_q},d_k\}}$ or $W_{\mathcal{B}\cup\{d_{u_q},i\}}$, and cached by either user $u_{g(d_k)}$ or user $u_{g(i)}$.

  By Lemma 1.Item 2, user k can recover the desired sub-block cached by user $u_{g(d_k)}$ from the first sub-phase. Each sub-block of $W_{\mathcal{B}\cup\{d_{u_q},d_k\}}$ cached by user $u_{g(i)}$ and not by $u_{g(d_k)}$ (assumed to be $W_{\mathcal{B}\cup\{d_{u_q},d_k\},\mathcal{V}'}$), can be recovered by user k from $C_{\mathcal{V}'\cup\{k\},\mathcal{B}\cup\{d_{u_q}\}}$ (transmitted in Step $g(i)$ of the first sub-phase), because all sub-blocks in $C_{\mathcal{V}'\cup\{k\},\mathcal{B}\cup\{d_{u_q}\}}$ except $W_{\mathcal{B}\cup\{d_{u_q},d_k\},\mathcal{V}'}$ are cached by user k. Hence, in $C_{\mathcal{J}\cup\{u_{g(i)}\}\setminus\{k_3\},\mathcal{B}\cup\{d_{u_q}\}}$, user k can recover all sub-blocks of $W_{\mathcal{B}\cup\{d_{u_q},d_k\}}$. So user k can recover the sum of the sub-blocks of $W_{\mathcal{B}\cup\{d_{u_q},i\}}$ in $C_{\mathcal{J}\cup\{u_{g(i)}\}\setminus\{k_3\},\mathcal{B}\cup\{d_{u_q}\}}$ from the first sub-phase,

$$I(k_3) := \bigoplus_{k_4 \in \mathcal{J}\cup\{u_{g(i)}\}\setminus\{k_3\}: d_{k_4} \neq d_k} W_{\mathcal{B}\cup\{d_{u_q},i\},\mathcal{J}\cup\{u_{g(i)}\}\setminus\{k_3,k_4\}}. \tag{65}$$

By the similar proof as (57) and (58), we can prove that

$$I \oplus \bigoplus_{k_3 \in \mathcal{J}'\cup\{u_q\}: d_{k_3} \neq d_k} I(k_3) = 0, \tag{66}$$

from the fact that each sub-block in (66) appears twice in (66). Hence, user k can recover I from the transmission of the first sub-phase. In addition, by the definition, we have

$$C_{\mathcal{J},\mathcal{B}} = C_{\mathcal{J},\mathcal{B}\cup\{i\}} \oplus C_{\mathcal{J},\mathcal{B}\cup\{d_{u_q}\}} \oplus I, \tag{67}$$

where $C_{\mathcal{J},\mathcal{B}\cup\{i\}}$ and $C_{\mathcal{J}',\mathcal{B}\cup\{d_{u_q}\}}$ are transmitted in Step $g(d_k)$ of the first sub-phase. Hence, user k can reconstruct $C_{\mathcal{J},\mathcal{B}}$ from the transmission of the first sub-phase.

In conclusion, we proved that from the transmission of the first sub-phase, user k can reconstruct $C_{\mathcal{J},\mathcal{B}}$.

Hence, from Observation 2, user k can recover $W_{\mathcal{S},\mathcal{V}}$ where $\mathcal{N}_{\mathbf{d}}(\mathcal{V}) \cap \{d_{u_1}, \ldots, d_{u_{g(d_k)-1}}\} = \emptyset$, from the transmission of the first sub-phase.

B. $\mathcal{N}_{\mathbf{d}}(\mathcal{V}) \cap \{d_{u_1}, \ldots, d_{u_{g(d_k)-1}}\} \neq \emptyset$

For the case $t = 2$, since $\mathcal{N}_{\mathbf{d}}(\mathcal{V}) \cap (\mathcal{S} \setminus \{d_k\}) \neq \emptyset$, $\mathcal{N}_{\mathbf{d}}(\mathcal{V}) \cap \{d_{u_1}, \ldots, d_{u_{g(d_k)-1}}\} \neq \emptyset$, $\{d_{u_1}, \ldots, d_{u_{g(d_k)-1}}\} \cap \mathcal{S} =$

Authorized licensed use limited to: University of Illinois at Chicago Library. Downloaded on November 08,2023 at 23:24:46 UTC from IEEE Xplore. Restrictions apply.

$\emptyset$, and $|\mathcal{V}|=2$, it can be seen that $|\mathcal{N}_{\mathbf{d}}(\mathcal{V})\setminus\mathcal{S}|=1$. For the case $\mathsf{r}=\mathsf{N}-1$, since $\mathcal{N}_{\mathbf{d}}(\mathcal{V})\cap\{d_{u_1},\ldots,d_{u_{g(d_k)-1}}\}\neq\emptyset$, $\{d_{u_1},\ldots,d_{u_{g(d_k)-1}}\}\cap\mathcal{S}=\emptyset$, and $|\mathcal{S}|=\mathsf{r}=\mathsf{N}-1$, it can also be seen that $|\mathcal{N}_{\mathbf{d}}(\mathcal{V})\setminus\mathcal{S}|=1$. In addition, in both two cases, since $d_k\in\mathcal{S}$, we have $d_k\notin(\mathcal{N}_{\mathbf{d}}(\mathcal{V})\setminus\mathcal{S})$. Hence, when $t=2$ or $\mathsf{r}=\mathsf{N}-1$, the interferences in $C_{\mathcal{V}\cup\{u_{g(d_k)}\},\mathcal{S}\setminus\{d_k\}}$ (transmitted in Step $g(d_k)$ of the first sub-phase) to user k are all from the block $W_{\mathcal{S}\cup\{i\}\setminus\{d_k\}}$, where we assume that i is the element in $\mathcal{N}_{\mathbf{d}}(\mathcal{V})\setminus\mathcal{S}$. The sum of the interferences in $C_{\mathcal{V}\cup\{u_{g(d_k)}\},\mathcal{S}\setminus\{d_k\}}$ to user k is

$$I'=\underset{k'\in\mathcal{V}:d_{k'}\neq d_k}{\oplus}W_{\mathcal{S}\cup\{i\}\setminus\{d_k\},\mathcal{V}\cup\{u_{g(d_k)}\}\setminus\{k'\}}. \tag{68}$$

For each user $k_1\in\mathcal{V}$ where $d_{k_1}\neq d_k$, we focus on $C_{\mathcal{V}\setminus\{k_1\}\cup\{u_{g(d_k)},u_{g(i)}\},\mathcal{S}\setminus\{d_k\}}$ which is transmitted in Step $g(i)$ of the first sub-phase. In $C_{\mathcal{V}\setminus\{k_1\}\cup\{u_{g(d_k)},u_{g(i)}\},\mathcal{S}\setminus\{d_k\}}$, since $|\mathcal{V}|=2$ (for the case $t=2$) or $|\mathcal{S}|=\mathsf{N}-1$ (for the case $\mathsf{r}=\mathsf{N}-1$), it can be seen that all sub-blocks are from either $W_{\mathcal{S}}$ or $W_{\mathcal{S}\setminus\{d_k\}\cup\{i\}}$. Each sub-block from $W_{\mathcal{S}}$ is cached by either user $u_{g(d_k)}$ or user $u_{g(i)}$, which can be recovered by user k from the first sub-phase, by Lemma 1.Item 2. Hence, user k can recover the sum of the sub-blocks of $W_{\mathcal{S}\setminus\{d_k\}\cup\{i\}}$ in $C_{\mathcal{V}\setminus\{k_1\}\cup\{u_{g(d_k)},u_{g(i)}\},\mathcal{S}\setminus\{d_k\}}$ as follows,

$$I'(k_1):=\underset{k'\in\mathcal{V}\setminus\{k_1\}\cup\{u_{g(i)}\}:d_{k'}\neq d_k}{\oplus}W_{\mathcal{S}\setminus\{d_k\}\cup\{i\},\mathcal{V}\setminus\{k_1,k'\}\cup\{u_{g(d_k)},u_{g(i)}\}}. \tag{69}$$

By the similar proof as (57) and (58), we can prove that

$$I'\oplus\underset{k_1\in\mathcal{V}:d_{k_1}\neq d_k}{\oplus}I'(k_1)=0, \tag{70}$$

from the fact that each sub-block in (70) appears twice in (70). Hence, user k can reconstruct the sum of all interferences I' in $C_{\mathcal{V}\cup\{u_{g(d_k)}\},\mathcal{S}\setminus\{d_k\}}$. Other sub-blocks in $C_{\mathcal{V}\cup\{u_{g(d_k)}\},\mathcal{S}\setminus\{d_k\}}$ are from the block $W_{\mathcal{S}}$ which is desired by user k. In addition, all these sub-blocks are cached by user $u_{g(d_k)}$ except $W_{\mathcal{S},\mathcal{V}}$. By Lemma 1.Item 2, from the transmission of first sub-phase user k can recover the sub-blocks of $W_{\mathcal{S}}$ which are cached by user $u_{g(d_k)}$. Hence, user k can also recover $W_{\mathcal{S},\mathcal{V}}$ in the first sub-phase.

Appendix G
Codes for Extension to Caching With Multiple Requests

For the coded caching problem with multiple requests considered in [35] where each user demands L independent and equal-length files, the proposed delivery scheme in [35] was proved to be optimal under the constraint of the MAN placement for most demands with $\mathsf{K}\leq 4$, $\mathsf{M}=\mathsf{N}/\mathsf{K}$, and $\mathsf{L}=2$, except one demand for $\mathsf{K}=3$ and three demands for $\mathsf{K}=4$. Different from the considered problem in this paper, the demands are not generally symmetric for the coded caching problem with multiple requests. Hence, for the coded caching problem with multiple requests, we pick a set of leaders such that each leader has at least one specific demanded file which is not demanded by other leaders, and the union set of demanded files by the leaders should be equal to the union set of demanded files by all users. In addition, the number of leaders should be as small as possible. We can then extend the proposed scheme for $t=1$ in order to achieve the optimality for those four exceptional demands, by satisfying the demands of leaders subsequently and aligning the interferences to non-leaders simultaneously.

1) $d_1=\{F_1,F_2\}$, $d_2=\{F_1,F_3\}$, and $d_3=\{F_2,F_3\}$ (case $\mathbf{D}_7$ in [35]). We use the MAN placement and divide each file F_i where $i\in[\mathsf{N}]$ into $\binom{\mathsf{K}}{t}$ non-overlapping and equal-length subfiles, $F_i=\{F_{i,\mathcal{W}}:\mathcal{W}\subseteq[\mathsf{K}],|\mathcal{W}|=t\}$, where $t=\mathsf{KM}/\mathsf{N}=1$. It can be seen this case is equivalent to our considered $(\mathsf{N},\mathsf{K},\mathsf{M},\mathsf{r})=(3,3,1,2)$ shared-link caching problem with correlated files of combinatorial overlaps. Hence, we can directly use the proposed delivery phase in this paper to transmit the linear combinations (with the permutation of leaders $(u_1,u_2)=(1,2)$)

Step 1: $F_{1,\{2\}}\oplus F_{1,\{1\}}$, $F_{1,\{3\}}\oplus F_{3,\{1\}}$, $F_{2,\{2\}}\oplus F_{3,\{1\}}$, $F_{2,\{3\}}\oplus F_{2,\{1\}}$;
Step 2: $F_{3,\{3\}}\oplus F_{3,\{2\}}$.

Hence, the load is $5/3$ which coincides with the converse bound under the constraint of MAN placement in [35], while the proposed caching scheme in [35] achieves 2.

2) $d_1=\{F_1,F_2\}$, $d_2=\{F_1,F_3\}$, $d_3=\{F_2,F_3\}$, and $d_4=\{F_4,F_5\}$ (case $\mathbf{D}'_{15}$ in [35]). It can be seen that if we only focus on the demands of users $1,2,3$, it is equivalent to our considered $(\mathsf{N},\mathsf{K},\mathsf{M},\mathsf{r})=(3,3,1,2)$ shared-link caching problem with correlated files of combinatorial overlaps. In addition, the demanded file by user 4 are independent of any demanded file by users $1,2,3$. Hence, we first satisfy the demands of user 4 and then use the codes for our considered $(\mathsf{N},\mathsf{K},\mathsf{M},\mathsf{r})=(3,3,1,2)$ shared-link caching problem with correlated files of combinatorial overlaps. Thus we transmit (with the permutation of leaders $(u_1,u_2,u_3)=(4,1,2)$)

Step 1: $F_{4,\{1\}}\oplus F_{1,\{4\}}$, $F_{4,\{2\}}\oplus F_{1,\{4\}}$, $F_{4,\{3\}}\oplus F_{3,\{4\}}$, $F_{5,\{1\}}\oplus F_{2,\{4\}}$, $F_{5,\{2\}}\oplus F_{3,\{4\}}$, $F_{5,\{3\}}\oplus F_{2,\{4\}}$;
Step 2: $F_{1,\{2\}}\oplus F_{1,\{1\}}$, $F_{1,\{3\}}\oplus F_{3,\{1\}}$, $F_{2,\{2\}}\oplus F_{3,\{1\}}$, $F_{2,\{3\}}\oplus F_{2,\{1\}}$;
Step 3: $F_{3,\{3\}}\oplus F_{3,\{2\}}$.

It can be checked that at the end of Step $j\in[3]$, each leader user u_j can recover its desired files by directly reading off. The non-leader (user 3) can recover $F_{2,\{1\}},F_{2,\{4\}},F_{3,\{1\}},F_{3,\{2\}},F_{3,\{4\}}$ by directly reading off, and recover $F_{2,\{2\}}$ by indirectly reading off. Hence, the load is $11/4$ which coincides with the converse bound under the constraint of MAN placement in [35], while the proposed caching scheme in [35] achieves 3.

3) $d_1=\{F_1,F_2\}$, $d_2=\{F_1,F_3\}$, $d_3=\{F_1,F_4\}$, and $d_4=\{F_2,F_3\}$ (case $\mathbf{D}'_{17}$ in [35]). We choose the permutation of leaders as $(3,4)$. Inspired from the proposed scheme for $t=1$, the delivery contains two steps where in the first and second steps, we satisfy the demands of users 3 and 4, respectively.
In Step 1, we first let user 3 recover F_1. For each user $k\in\{1,2,4\}$, if F_1 is demanded by user k, we transmit

Authorized licensed use limited to: University of Illinois at Chicago Library. Downloaded on November 08,2023 at 23:24:46 UTC from IEEE Xplore. Restrictions apply.

$F_{1,\{k\}} \oplus F_{1,\{3\}}$; otherwise, we pick one demanded file by user k which is not F_4 (assumed to be F_i), and transmit $F_{1,\{k\}} \oplus F_{i,\{3\}}$.

We then let user 3 recover F_4. For each user $k \in \{1,2,4\}$, if F_4 is demanded by user k, we transmit $F_{4,\{k\}} \oplus F_{4,\{3\}}$; otherwise, we pick one demanded file by user k which is not F_i nor F_1 (assumed to be $F_{i'}$), and transmit $F_{4,\{k\}} \oplus F_{i',\{3\}}$.

By this way, we transmit in the the steps (with the permutation of leaders $(u_1, u_2) = (3, 4)$)

$$\begin{aligned}\text{Step 1: } & F_{1,\{1\}} \oplus F_{1,\{3\}},\ F_{1,\{2\}} \oplus F_{1,\{3\}},\\ & F_{1,\{4\}} \oplus F_{3,\{3\}},\ F_{4,\{1\}} \oplus F_{2,\{3\}},\\ & F_{4,\{2\}} \oplus F_{3,\{3\}},\ F_{4,\{4\}} \oplus F_{2,\{3\}};\\ \text{Step 2: } & F_{3,\{1\}} \oplus F_{1,\{4\}},\ F_{3,\{2\}} \oplus F_{3,\{4\}},\\ & F_{2,\{1\}} \oplus F_{2,\{4\}},\ F_{2,\{2\}} \oplus F_{1,\{4\}}.\end{aligned}$$

It can be checked that at the end of Step $j \in [2]$, each leader user u_j can recover its desired files by directly reading off. For the non-leaders (users $1, 2$), user 1 can recover $F_{1,\{3\}}, F_{1,\{4\}}, F_{2,\{3\}}, F_{2,\{4\}}$ by directly reading off, and recover $F_{1,\{2\}}, F_{2,\{2\}}$ by indirectly reading off; user 2 can recover $F_{1,\{3\}}, F_{1,\{4\}}, F_{3,\{3\}}, F_{3,\{4\}}$ by directly reading off, and recover $F_{1,\{1\}}, F_{3,\{1\}}$ by indirectly reading off. Hence, the load is $10/4$ which coincides with the converse bound under the constraint of MAN placement in [35], while the proposed caching scheme in [35] achieves $11/4$.

4) $d_1 = \{F_1, F_2\}$, $d_2 = \{F_1, F_2\}$, $d_3 = \{F_1, F_3\}$, and $d_4 = \{F_2, F_3\}$ (case $\mathbf{D}'_{20}$ in [35]). It can be seen that this case is equivalent to our considered $(\mathsf{N}, \mathsf{K}, \mathsf{M}, \mathsf{r}) = (3, 4, 1, 2)$ shared-link caching problem with correlated files of combinatorial overlaps. Hence, we can directly use the proposed delivery phase in this paper to transmit the linear combinations (with the permutation of leaders $(u_1, u_2) = (1, 3)$)

$$\begin{aligned}\text{Step 1: } & F_{1,\{2\}} \oplus F_{1,\{1\}},\ F_{1,\{3\}} \oplus F_{1,\{1\}},\\ & F_{1,\{4\}} \oplus F_{3,\{1\}},\ F_{2,\{2\}} \oplus F_{2,\{1\}},\\ & F_{2,\{3\}} \oplus F_{3,\{1\}},\ F_{2,\{4\}} \oplus F_{2,\{1\}};\\ \text{Step 2: } & F_{3,\{2\}} \oplus F_{2,\{3\}},\ F_{3,\{4\}} \oplus F_{3,\{3\}}.\end{aligned}$$

Hence, the load is 2, which coincides with the converse bound under the constraint of MAN placement in [35], while the proposed caching scheme in [35] achieves $9/4$.

REFERENCES

[1] K. Wan, D. Tuninetti, M. Ji, and G. Caire, "On coded caching with correlated files," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jul. 2019, pp. 692–696.

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

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

[4] Q. Yu, M. A. Maddah-Ali, and A. S. Avestimehr, "The exact rate-memory tradeoff for caching with uncoded prefetching," *IEEE Trans. Inf. Theory*, vol. 64, no. 2, pp. 1281–1296, Feb. 2018.

[5] Q. Yu, M. A. Maddah-Ali, and A. S. Avestimehr, "Characterizing the rate-memory tradeoff in cache networks within a factor of 2," *IEEE Trans. Inf. Theory*, vol. 65, no. 1, pp. 647–663, Jan. 2019.

[6] M. A. Maddah-Ali and U. Niesen, "Decentralized coded caching attains order-optimal memory-rate tradeoff," *IEEE/ACM Trans. Netw.*, vol. 23, no. 4, pp. 1029–1040, Aug. 2015.

[7] M. Ji, G. Caire, and A. F. Molisch, "Fundamental limits of caching in wireless D2D networks," *IEEE Trans. Inf. Theory*, vol. 62, no. 2, pp. 849–869, Feb. 2016.

[8] N. Karamchandani, U. Niesen, M. A. Maddah-Ali, and S. N. Diggavi, "Hierarchical coded caching," *IEEE Trans. Inf. Theory*, vol. 62, no. 6, pp. 3212–3229, Jun. 2016.

[9] M. Ji, A. M. Tulino, J. Llorca, and G. Caire, "Caching in combination networks," in *Proc. 49th Asilomar Conf. Sig., Sys. Comp.*, Nov. 2015, pp. 1–12.

[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," *IEEE Trans. Inf. Theory*, vol. 68, no. 2, pp. 806–827, Feb. 2022.

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

[12] K. Wan, H. Sun, M. Ji, D. Tuninetti, and G. Caire, "Cache-aided general linear function retrieval," *Entropy*, vol. 23, no. 1, p. 25, Dec. 2020.

[13] K. Wan, H. Sun, M. Ji, D. Tuninetti, and G. Caire, "Cache-aided matrix multiplication retrieval," *IEEE Trans. Inf. Theory*, vol. 68, no. 7, pp. 4301–4319, Jul. 2022.

[14] A. Sengupta, R. Tandon, and T. C. Clancy, "Fundamental limits of caching with secure delivery," *IEEE Trans. Inf. Forensics Security*, vol. 10, no. 2, pp. 355–370, Feb. 2015.

[15] K. Wan and G. Caire, "On coded caching with private demands," *IEEE Trans. Inf. Theory*, vol. 67, no. 1, pp. 358–372, Jan. 2021.

[16] Q. Yan and D. Tuninetti, "Fundamental limits of caching for demand privacy against colluding users," *IEEE J. Sel. Areas Inf. Theory*, vol. 2, no. 1, pp. 192–207, Mar. 2021.

[17] P. Hassanzadeh, A. Tulino, J. Llorca, and E. Erkip, "Correlation-aware distributed caching and coded delivery," in *Proc. IEEE Inf. Theory Workshop (ITW)*, Sep. 2016, pp. 166–170.

[18] A. El Gamal and Y.-H. Kim, *Network Information Theory*. Cambridge, U.K.: Cambridge Univ. Press, 2011.

[19] P. Hassanzadeh, A. M. Tulino, J. Llorca, and E. Erkip, "Rate-memory trade-off for caching and delivery of correlated sources," *IEEE Trans. Inf. Theory*, vol. 66, no. 4, pp. 2219–2251, Apr. 2020.

[20] R. M. Gray and A. D. Wyner, "Source coding for a simple network," *Bell Syst. Tech. J.*, vol. 53, no. 9, pp. 1681–1721, Nov. 1974.

[21] Q. Yang and D. Gunduz, "Centralized coded caching of correlated contents," in *Proc. IEEE Int. Conf. Commun.*, May 2018, pp. 1–6.

[22] M. Ji et al., "On the fundamental limits of caching in combination networks," in *Proc. IEEE 16th Int. Workshop Signal Process. Adv. Wireless Commun. (SPAWC)*, Jun. 2015, pp. 695–699.

[23] L. Tang and A. Ramamoorthy, "Coded caching for networks with the resolvability property," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jul. 2016, pp. 420–424.

[24] A. A. Zewail and A. Yener, "Combination networks with or without secrecy constraints: The impact of caching relays," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 6, pp. 1140–1152, Jun. 2018.

[25] Q. Yan, M. Wigger, and S. Yang, "Placement delivery array design for combination networks with edge caching," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jun. 2018, pp. 1555–1559.

[26] P. N. Muralidhar, D. Katyal, and B. S. Rajan, "Maddah-Ali-Niesen scheme for multi-access coded caching," in *Proc. IEEE Inf. Theory Workshop (ITW)*, Oct. 2021, pp. 1–6.

[27] F. Brunero and P. Elia, "Fundamental limits of combinatorial multi-access caching," *IEEE Trans. Inf. Theory*, vol. 69, no. 2, pp. 1037–1056, Feb. 2023.

[28] F. Brunero and P. Elia, "Unselfish coded caching can yield unbounded gains over selfish caching," *IEEE Trans. Inf. Theory*, vol. 68, no. 12, pp. 7871–7891, Dec. 2022.

[29] S. Li, M. A. Maddah-Ali, Q. Yu, and A. S. Avestimehr, "A fundamental tradeoff between computation and communication in distributed computing," *IEEE Trans. Inf. Theory*, vol. 64, no. 1, pp. 109–128, Jan. 2018.

[30] P. Hassanzadeh, A. M. Tulino, J. Llorca, and E. Erkip, "On coding for cache-aided delivery of dynamic correlated content," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 8, pp. 1666–1681, Aug. 2018.

[31] Q. Yang, P. Hassanzadeh, D. Gunduz, and E. Erkip, "Centralized caching and delivery of correlated contents over a Gaussian broadcast channel," *IEEE Trans. Commun.*, vol. 68, no. 1, pp. 122–136, Jan. 2020.

Authorized licensed use limited to: University of Illinois at Chicago Library. Downloaded on November 08,2023 at 23:24:46 UTC from IEEE Xplore. Restrictions apply.

[32] M. Ji, A. M. Tulino, J. Llorca, and G. Caire, "Caching and coded multicasting: Multiple groupcast index coding," in *Proc. IEEE Global Conf. Signal Inf. Process. (GlobalSIP)*, Dec. 2014, pp. 881–885.

[33] M. Ji, A. Tulino, J. Llorca, and G. Caire, "Caching-aided coded multicasting with multiple random requests," in *Proc. IEEE Inf. Theory Workshop (ITW)*, Apr. 2015, pp. 1–5.

[34] A. Sengupta and R. Tandon, "Improved approximation of storage-rate tradeoff for caching with multiple demands," *IEEE Trans. Commun.*, vol. 65, no. 5, pp. 1940–1955, May 2017.

[35] Y. Wei and S. Ulukus, "Coded caching with multiple file requests," in *Proc. 55th Annu. Allerton Conf. Commun., Control, Comput. (Allerton)*, Oct. 2017, pp. 437–442.

[36] A. G. Sheshjavani, A. Khonsari, S. P. Shariatpanahi, M. Moradian, and A. Dadlani, "Coded caching under non-uniform content popularity distributions with multiple requests," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, May 2020, pp. 1–6.

[37] K. Huang, X. Cai, J. Zhang, and Z. Luo, "Coded caching with distinct number of user requests," in *Proc. IEEE Global Commun. Conf.*, Dec. 2020, pp. 1–6.

[38] E. Parrinello, A. Ünsal, and P. Elia, "Fundamental limits of coded caching with multiple antennas, shared caches and uncoded prefetching," *IEEE Trans. Inf. Theory*, vol. 66, no. 4, pp. 2252–2268, Apr. 2020.

[39] N. S. Karat, S. Dey, A. Thomas, and B. S. Rajan, "An optimal linear error correcting delivery scheme for coded caching with shared caches," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jul. 2019, pp. 1217–1221.

[40] K. Wan, D. Tuninetti, M. Ji, and G. Caire, "On the fundamental limits of fog-RAN cache-aided networks with downlink and sidelink communications," 2018, *arXiv:1811.05498*.

[41] H. Xu, C. Gong, and X. Wang, "Efficient file delivery for coded prefetching in shared cache networks with multiple requests per user," *IEEE Trans. Commun.*, vol. 67, no. 4, pp. 2849–2865, Apr. 2019.

[42] A. M. Ibrahim, A. A. Zewail, and A. Yener, "Coded placement for systems with shared caches," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2019, pp. 1–6.

[43] E. Peter and B. S. Rajan, "Multi-antenna coded caching from a placement delivery array for shared caches," *IEEE Trans. Commun.*, vol. 70, no. 6, pp. 3627–3640, Jun. 2022.

[44] E. Peter, K. K. K. Namboodiri, and B. S. Rajan, "Shared cache coded caching schemes with known user-to-cache association profile using placement delivery arrays," 2022, *arXiv:2201.10577*.

[45] Q. Yan, M. Cheng, X. Tang, and Q. Chen, "On the placement delivery array design for centralized coded caching scheme," *IEEE Trans. Inf. Theory*, vol. 63, no. 9, pp. 5821–5833, Sep. 2017.

[46] F. Arbabjolfaei, B. Bandemer, Y. Kim, E. Sasoglu, and L. Wang, "On the capacity region for index coding," in *Proc. IEEE Int. Symp. Inf. Theory*, Jul. 2013, pp. 962–966.

[47] Z. Bar-Yossef, Y. Birk, T. S. Jayram, and T. Kol, "Index coding with side information," *IEEE Trans. Inf. Theory*, vol. 57, no. 3, pp. 1479–1494, Mar. 2011.

Kai Wan (Member, IEEE) received the B.E. degree in optoelectronics from the Huazhong University of Science and Technology, China, in 2012, and the M.Sc. and Ph.D. degrees in communications from Université Paris-Saclay, France, in 2014 and 2018, respectively. Subsequently, he was a Post-Doctoral Researcher with the Communications and Information Theory Chair (CommIT), Technische Universität Berlin, Berlin, Germany. He is currently a Professor with the School of Electronic Information and Communications, Huazhong University of Science and Technology. His research interests include information theory, coding techniques and their applications on coded caching, index coding, distributed storage, distributed computing, wireless communications, and privacy and security. He has served as an Associate Editor for IEEE COMMUNICATIONS LETTERS in August 2021.

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 in Sophia Antipolis, France). She is currently a Professor with the Department of Electrical and Computer Engineering, 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 ultimate performance limits of wireless interference networks (with a 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 University of Illinois Scholar in 2015. She was the Editor-in-Chief of the IEEE INFORMATION THEORY SOCIETY NEWSLETTER, from 2006 to 2008; and an Editor of IEEE COMMUNICATION 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 also currently an Editor of IEEE TRANSACTIONS ON COMMUNICATIONS. She is currently a Distinguished Lecturer of the Information Theory Society.

Mingyue Ji (Member, IEEE) received the B.E. degree in communication engineering from the Beijing University of Posts and Telecommunications, China, in 2006, the M.Sc. degrees in electrical engineering from the Royal Institute of Technology, Sweden, and the University of California, Santa Cruz, in 2008 and 2010, respectively, and the Ph.D. degree from the Ming Hsieh Department of Electrical Engineering, University of Southern California, in 2015. Subsequently, he was a Staff II System Design Scientist with Broadcom Corporation (Broadcom Ltd.), from 2015 to 2016. He is currently an Associate Professor with the Electrical and Computer Engineering Department and an Adjunct Associate Professor with the School of Computing, The University of Utah. His research interests include information theory, coding theory, concentration of measure and statistics, with the applications of distributed computing systems, wireless communications and networking, caching networks, distributed machine learning, distributed storage, and (statistical) signal processing. He has received the NSF CAREER Award in 2022, the IEEE Communications Society Leonard G. Abraham Prize for the Best IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS Paper in 2019, the Best Paper Awards at 2021 IEEE GLOBECOM Conference and 2015 IEEE ICC Conference, the Best Student Paper Award at 2010 IEEE European Wireless Conference, and the USC Annenberg Fellowship from 2010 to 2014. He has been serving as an Associate Editor for IEEE TRANSACTIONS ON INFORMATION THEORY, since 2022, and IEEE TRANSACTIONS ON COMMUNICATIONS, since 2020.

Giuseppe Caire (Fellow, IEEE) was born in Torino in 1965. He received the B.Sc. degree in electrical engineering from Politecnico di Torino in 1990, the M.Sc. degree in electrical engineering from Princeton University in 1992, and the Ph.D. degree from Politecnico di Torino in 1994. He was a Post-Doctoral Research Fellow with the European Space Agency (ESTEC), Noordwijk, The Netherlands, from 1994 to 1995; an Assistant Professor of telecommunications with Politecnico di Torino; an Associate Professor with the University of Parma, Italy; a Professor with the Department of Mobile Communications, Eurecom Institute, Sophia-Antipolis, France; and a Professor of electrical engineering with the Viterbi School of Engineering, University of Southern California, Los Angeles. He is currently an Alexander von Humboldt Professor with the Faculty of Electrical Engineering and Computer Science, Technische Universität Berlin, Germany. His main research interests include communications theory, information theory, and channel and source coding, with particular focus on wireless communications. He received the Jack Neubauer Best System Paper Award from the IEEE Vehicular Technology Society in 2003, the IEEE Communications Society and Information Theory Society Joint Paper Award in 2004 and 2011, the Okawa Research Award in 2006, the Alexander von Humboldt Professorship in 2014, the Vodafone Innovation Prize in 2015, an ERC Advanced Grant in 2018, the Leonard G. Abraham Prize for Best IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS Paper in 2019, and the IEEE Communications Society Edwin Howard Armstrong Achievement Award in 2020. He was a recipient of the 2021 Leibinz Prize of the German National Science Foundation (DFG). He has served in the Board of Governors of the IEEE Information Theory Society, from 2004 to 2007, where he was an Officer, from 2008 to 2013. He was a President of the IEEE Information Theory Society in 2011.

Authorized licensed use limited to: University of Illinois at Chicago Library. Downloaded on November 08,2023 at 23:24:46 UTC from IEEE Xplore. Restrictions apply.