

Full Coded Caching Gains for Cache-less Users

Eleftherios Lampiris and Petros Elia

Abstract— Within the context of coded caching, the work reveals the interesting connection between having multiple transmitters and having heterogeneity in the cache sizes of the receivers. Our work effectively shows that having multiple transmit antennas – while providing full multiplexing gains – can also simultaneously completely remove the performance penalties that are typically associated to cache-size unevenness. Focusing on the multiple-input single-output Broadcast Channel, the work first identifies the performance limits of the extreme case where cache-aided users coincide with users that do not have caches, and then expands the analysis to the case where both user groups are cache-aided but with heterogeneous cache-sizes. In the first case, the main contribution is a new algorithm that employs perfect matchings on a bipartite graph to offer full multiplexing as well as *full* coded-caching gains to *both* cache-aided as well as cache-less users. An interesting conclusion is that, starting from a single-stream centralized coded caching setting with normalized cache size γ , then adding L antennas allows for the addition of approximately L/γ extra *cache-less* users, at no added delay costs. Similarly surprising is the finding that, starting off with a single-antenna hybrid system (with both cache-less and cache-aided users), then adding $L-1$ antennas to the transmitter, as well as endowing the cache-less users with a cumulative normalized cache size Γ_2 , increases the Degrees of Freedom by a *multiplicative* factor of up to $\Gamma_2 + L$.

Index Terms—Caching networks, coded caching, heterogeneous cache sizes, delivery rate, uncoded cache placement, index coding, MISO broadcast channel, network coding.

I. INTRODUCTION

Coded caching is a technique — first introduced in [1] for the single-stream bottleneck broadcast channel (BC) — that exploits receiver-side caches in order to deliver cacheable content to many users at a time. This technique initially involved a setting where a single-antenna transmitter has access to a library of N files, and serves (via a single bottleneck link) K receivers, each having a cache of size equal to the size of M files. The process involved a novel cache placement method and a subsequent delivery phase during which each user simultaneously requests one library file, while the transmitter employs cache-dependent coded multicasting to simultaneously deliver independent requested content to many users at a time.

In the single stream setting ($L = 1$ transmit antenna), where the bottleneck link has capacity equal to 1 file per unit of time,

Eleftherios Lampiris is with the Electrical Engineering and Computer Science Department, Technische Universität Berlin, 10587 Berlin, Germany (lampiris@tu-berlin.de) and Petros Elia is with the Communication Systems Department at EURECOM, Sophia Antipolis, 06410, France (elia@eurecom.fr). This work was conducted while Eleftherios was employed by EURECOM. The work is supported by the European Research Council under the EU Horizon 2020 research and innovation program / ERC grant agreement no. 725929. (ERC project DUALITY). Part of this work was presented in ITW 2018 [2].

the work in [1] showed that any set of K simultaneous requests can be served with *normalized* delay (worst-case completion time, guaranteeing the delivery of any set of requested files) which is at most

$$T_{L=1}(K, \gamma) = \frac{K(1 - \gamma)}{1 + K\gamma}, \quad (1)$$

where $\gamma \triangleq \frac{M}{N} \in [0, 1)$ denotes the normalized cache size. This implied the ability to treat $K\gamma + 1$ cache-aided users at a time; a number that is often referred to as the cache-aided sum *Degrees of Freedom* (DoF)

$$D_{L=1}(K, \gamma) \triangleq \frac{K(1 - \gamma)}{T_1(K, \gamma)} = 1 + K\gamma, \quad (2)$$

corresponding to a caching gain of $K\gamma$ additional served users due to caching.

A. Multi-antenna coded caching

Recently, coded caching has been explored in the presence of multiple antennas/transmitters. In the context of a fully-connected multiple-input single-output (MISO) BC, multi-antenna (L antennas) techniques were combined with coded caching to reveal new insights such as that i) multiplexing and caching gains can be combined additively [3], [4] to yield a sum-DoF of $D_L(K, \gamma) = L + K\gamma$, ii) multiple antennas can dramatically reduce subpacketization, thus allowing for multiplicative DoF gains in the finite file-size regime [5], that iii) the feedback cost of combining the two gains need only be a function of the number of antennas and not of the caching gain [6], [7], iv) in the finite Signal-to-Noise-Ratio (SNR), choosing a smaller amount of transmitted streams can increase the performance [8]–[10] as well as other insights (cf. [11]–[21], etc.).

B. Coded caching with heterogeneous cache sizes

While the first works on coded caching focused more on the setting where users have identically sized caches, in reality many communication systems may include users with heterogeneous caching capabilities. It is expected, for example, that users with different types of devices, such as laptops and mobile phones which, naturally, have different storage constraints, will be simultaneously active. Moreover, different users may well decide to allocate different amounts of their available storage (or none of it) for content caching.

These uneven storage constraints can conceivably hamper the performance of coded caching systems. For example, imagining a system that needs to treat cache-aided and cache-less users, we can see that when users request different content, transmitting coded messages to cache-aided users can preclude the cache-less users from receiving any valuable information.

The expectation that users with uneven cache constraints might co-exist, has sparked a number of recent works that sought to ameliorate the effects of cache size unevenness [22]–[29]. For example, the work in [22] adopts the approach of splitting the caches into multiple layers while caching at each layer according to the algorithm of [1], and adjusted to the users’ cache sizes and to the size allocated to each cache layer. Further, the work in [23] considered the uneven cache-size scenario under decentralized placement¹. Moreover, the work in [26] explores the Device-to-Device setting with heterogeneous caches, while [24] investigates the fundamental limits of the single-stream Coded Caching with 2 users and uneven cache sizes. The works in [27]–[29] view the heterogeneous cache problem as a set of optimization constraints, where the size of each conceivable subfile is optimized in order to reduce the transmission time. Another idea can be found in [30] which adopts a coded placement approach in order to further increase the coded caching gains in the heterogeneous cache setting.

While for the homogeneous case we know that the scheme in [1] is optimal under the assumption of uncoded placement [31] (and approximately optimal [32] under any placement scheme), the optimal performance of the heterogeneous cache-size setting is not known in general. What is known though, from all the above works, is that cache-size heterogeneity has always entailed performance penalties compared to the homogeneous case.

C. Current setting, and brief summary of contributions

In the current setting, we will study the role of multiple antennas in tackling the penalties associated with having heterogeneous caches. Specifically, we will first consider a system where a set of K_1 users are each assisted by caches of some normalized size $\gamma_1 > 0$, while the remaining $K_2 = K - K_1$ users are cache-less ($\gamma_2 = 0$). In the single-antenna case we will show that, under the assumption of uncoded placement, the *optimal strategy* is to treat each set separately, thus revealing that a single-stream system is severely penalized by the presence of cache-less users.

Motivated by the above, we will then shift our focus to the study of the multiple antenna case (L antennas), where we will show that for a wide range of parameters, we are able to simultaneously treat both user types with DoF $D_L(K_1, \gamma_1, K_2, \gamma_2) = K_1\gamma_1 + L$ that will be equal to that of the corresponding homogeneous setting. Moreover, for the other case, where the DoF of the homogeneous system cannot be achieved, we will show that the DoF performance is L times higher than in the single antenna case, and we will show that this performance is *exactly optimal* under the assumption of uncoded cache placement.

We will then proceed to explore how the performance is boosted when now the $K_2 = K - K_1$ users of the second group are each endowed with a cache of normalized size

¹The main idea behind decentralized placement is to circumvent the fact that the identity of users needs to be known during the placement phase. Thus, storing of content at the users migrates from the deterministic placement introduced in [1] to a random placement.

$\gamma_2 \in (0, \gamma_1)$. First, we will prove that the total DoF of $L + K_1\gamma_1 + K_2\gamma_2$ can be achieved for a broad range of parameters. Further, for the case when this gain cannot be achieved, we will show that the same performance boost experienced in the cache-less case when adding $L - 1$ antennas (by a multiplicative factor of L) can be also achieved by adding caches to the cache-less group. Specifically, starting from the single antenna setting with the cache-aided and cache-less user sets, then adding a cumulative cache equal to Γ_2 at the cache-less group and $L - 1$ antennas we can achieve a *multiplicative* DoF boost by a factor of up to $\Gamma_2 + L$. The above results will reveal the power of multiple antennas in significantly or entirely removing the negative effects of cache-size unevenness, as well as the powerful effect that modest amounts of caching can have in uneven scenarios.

Again we stress that the above binary scenarios have particular practical pertinence. In the first scenario, the cache-less users may reflect users that employ legacy devices that do not support cache-aided decoding or that may wish to opt-out of dedicating their storage for caching (cf. [33]). The second scenario with two distinct cache sizes γ_1, γ_2 ($\gamma_1 > \gamma_2 > 0$) reflects the expectation that users are split between those that have laptop devices that can store more information, and those with mobile devices which generally have more serious storage constraints.

System Model

The goal of this work is to study the DoF performance of the L -antenna MISO BC² with K single-antenna receiving users, which are equipped with caches of heterogeneous sizes. Specifically, we focus on a system where a user belongs to one of two sets; the K_1 users of set \mathcal{K}_1 are endowed with caches of normalized size $\gamma_1 \in (0, 1)$, while each of the remaining $K_2 = K - K_1$ users of set \mathcal{K}_2 have caches of normalized size $\gamma_2 \in [0, \gamma_1)$. Each user simultaneously asks for a single – different – file, from a library of $N \geq K$ files, thus the metric of interest is the worst-case delivery time. In order to satisfy the users’ demands, the base station transmits an $L \times 1$ vector \mathbf{x} . Then the signal at each receiver k takes the form

$$y_k = \mathbf{h}_k^H \mathbf{x} + w_k, \quad k \in \{1, 2, \dots, K\} \triangleq [K]$$

where $\mathbf{h}_k \in \mathbb{C}^{L \times 1}$ denotes the channel between the transmitter and receiver k , where \mathbf{x} satisfies the power constraint $\mathbb{E}\{\|\mathbf{x}\|^2\} = P$, and where $w_k \sim \mathcal{CN}(0, 1)$ corresponds to the noise observed at user k . We assume that each node has all necessary channel-state information, and that for a given SNR, each link has capacity of the form $\log(\text{SNR}) + o(\log(\text{SNR}))$.

²We note that, while here we focus on the MISO BC, the results can be readily extended to the wired multiserver setting of [3], where a set of L servers are connected to K users and where the transmitted messages are linearly combined to form a full rank matrix between servers and users. The results can also be readily extended to the multiple transmitter interference setting where each of the K_T transmitters stores fraction γ_T of the content, such that $K_T\gamma_T = L$ (cf. [4]). Thus, all our results can be trivially applied in the multiple-transmitter/multi-server settings.

Our aim is to design, for the heterogeneous system at hand, a pre-fetching and delivery algorithm that minimizes the worst-case completion time

$$T_L(K_1, \gamma_1, K_2, \gamma_2) \quad (3)$$

corresponding to each user asking for a different file.

II. MAIN RESULTS

We begin with the case where users of set \mathcal{K}_2 are cache-less ($\gamma_2 = 0$), and then we will generalize by treating the case where $\gamma_2 \in (0, \gamma_1)$.

A. Coexistence of cache-aided and cache-less users

We start with a result that exemplifies — in the single stream case of $L = 1$ — the problem with having cache-aided users coexisting with cache-less users. We will use notation

$$T_{K_i} \triangleq \frac{K_i(1 - \gamma_i)}{1 + K_i\gamma_i} \quad (4)$$

to describe the delay needed to serve, in the single antenna setting, K_i cache-aided users with caches of normalized size γ_i (in the absence of any cache-less users) using 1 transmit antenna, where this performance is exactly optimal under the assumption of uncoded cache placement.

Theorem 1. *In a single-stream BC with K_1 cache-aided users equipped with caches of normalized size γ_1 and with K_2 additional cache-less users, the optimal delay, under the assumption of uncoded placement, takes the form*

$$T_1(K_1, \gamma_1, K_2, \gamma_2 = 0) = \frac{K_1(1 - \gamma_1)}{1 + K_1\gamma_1} + K_2. \quad (5)$$

Proof. The proof is relegated to Appendix A. \square

The above reveals that in the single stream case, every time a single cache-less user is added, there is a delay penalty of an entire unit of time, thus revealing that the two types of users should be treated separately. If such separation were to be applied in the multi-antenna case, the achievable performance would be

$$T_L(K_1, \gamma_1, K_2, \gamma_2 = 0) = \frac{K_1(1 - \gamma_1)}{L + K_1\gamma_1} + \frac{K_2}{L} \quad (6)$$

so the K_2 cache-less users would experience only a multiplexing gain of L , and would experience no caching gain.

We proceed with the main result of this work.

Theorem 2. *In the MISO BC with $L \geq 1$ antennas, K_1 cache-aided users equipped with cache of fractional size γ_1 , and $K_2 \geq (L - 1)T_{K_1}$ cache-less users, the delivery time*

$$T_L(K_1, \gamma_1, K_2, \gamma_2 = 0) = T_{K_1} + \frac{K_2 - (L - 1)T_{K_1}}{\min\{L, K_2\}} \quad (7)$$

is achievable and within a factor of 2 from optimal, while if $K_2 \leq (L - 1)T_{K_1}$ then

$$T_L(K_1, \gamma_1, K_2, \gamma_2 = 0) = \frac{K_2 + K_1(1 - \gamma_1)}{K_1\gamma_1 + L} \quad (8)$$

is achievable and within a factor of 3 from optimal under the assumption of linear and one-shot schemes.

Proof. The achievability part of the proof can be found in Section III, while the outer bound and gap calculations can be found in Appendix B. \square

Furthermore, the following theorem accentuates the multiplicative and optimal nature of the gains from adding antennas.

Theorem 3. *Starting from the single-antenna BC with K_1 cache-aided users with caches of normalized size γ_1 and $K_2 = (\tilde{L} - 1)T_{K_1}$ cache-less users (for any positive \tilde{L}), then going from 1 to $L \leq \tilde{L}$ antennas, reduces delay by L times, to a delay that is optimal under the assumption of uncoded cache placement.*

Proof. The calculation of the performance comes directly from Theorems 1 and 2, while the proof of optimality of the scheme is proved in Appendix C. \square

Let us proceed with a few corollaries that explore some of the ramifications of the above theorem. Eq. (6) helps us place the following corollary into context.

Corollary 1. *In the L -antenna, $(K_1, \gamma_1, K_2, \gamma_2 = 0)$ MISO BC with $K_2 \leq (L - 1)T_{K_1}$, all cache-aided and cache-less users can experience full multiplexing gain L as well as full caching gain $K_1\gamma_1$.*

Proof. The proof is direct from Eq. (8). \square

Example 1. *Let us assume the setting with $K_2 = 2$ cache-less users, and with $K_1 = 5$ cache-aided users each equipped with a cache of normalized size $\gamma_1 = \frac{1}{5}$. Transmitting with one antenna can achieve an optimal delay, under the assumption of uncoded placement, of*

$$T_1\left(5, \frac{1}{5}, 2, 0\right) = \frac{K_1(1 - \gamma_1)}{K_1\gamma_1 + 1} + K_2 = 4. \quad (9)$$

Going from $L = 1$ to $L = 2$ antennas, reduces the delay by a factor of 2, to the delay

$$T_2\left(5, \frac{1}{5}, 2, 0\right) = \frac{K_1(1 - \gamma_1) + K_2}{K_1\gamma_1 + 2} = 2 \quad (10)$$

which is optimal under the assumption of uncoded cache placement.

We quickly note that the above multiplicative boost of the DoF is in contrast to the additive DoF boost (additive multiplexing gain) experienced in systems with only cache-aided users [3]. This further advocates for the use of multiple transmit antennas in heterogeneous cache systems.

We proceed with another corollary which can be placed into context, by noting that in a system with L antennas and K_2 cache-less users, adding one more antenna would allow (without added delay costs) the addition of only a diminishing number of $\frac{K_2}{L}$ extra cache-less users.

Corollary 2. *Let us start from the single-stream BC with K_1 cache-aided users equipped with caches of normalized size γ_1 .*

Then, adding an extra $L - 1$ transmit antennas, allows for the addition of

$$K_2 = (L - 1)T_{K_1} \approx \frac{L - 1}{\gamma_1} \quad (11)$$

cache-less users, at no added delay costs.

Proof. This is direct from Theorem 2. \square

The following takes another point of view and explores the benefits of injecting cache-aided users into legacy (cache-less) MISO BC systems. To put the following corollary into context, we recall that in a cache-less L transmit-antenna MISO BC serving $K_2 \geq L$ users, the optimal (normalized) delay is $\frac{K_2}{L}$.

Corollary 3. *In a MISO BC with $K_2 \geq L$ cache-less users, introducing K_1 additional cache-aided users with $\gamma_1 \geq \frac{L}{K_2}$, incurs delay*

$$T_L(K_1, \gamma_1, K_2, \gamma_2 = 0) \leq \frac{K_2}{L - 1}$$

and thus we can add an infinite number of cache-aided users and only suffer a delay increase by a factor that is at most $\frac{L}{L-1}$.

Proof. This is direct from Theorem 2. \square

Multiple antennas for ‘balancing’ cache-size unevenness: In the variety of works (cf. Sec. I-B) that explore the single-stream coded caching setting in the presence of uneven cache sizes, we see that having cache-size asymmetry induces delay penalties and that the preferred cache-size allocation is the uniform one. The following corollary addresses this issue, in the multi-antenna setting.

Corollary 4. *The L -antenna, $(K_1, \gamma_1, K_2, \gamma_2 = 0)$ MISO BC with $K_2 \leq (L - 1)T_{K_1}$ cache-less users, incurs the same achievable delay*

$$T_L(K_1, \gamma_1, K_2, \gamma_2 = 0) = \frac{K_2 + K_1(1 - \gamma_1)}{L + K_1\gamma_1} = \frac{K(1 - \gamma_{av})}{L + K\gamma_{av}}$$

as the order optimal homogeneous K -user MISO BC with homogeneous caches of normalized size $\gamma_{av} = \frac{K_1\gamma_1}{K}$ (same cumulative cache size $K_1\gamma_1 = K\gamma_{av}$).

Proof. This is direct from Theorem 2. \square

Example 2. *Let us assume the $(K_1 = 5, \gamma_1 = 1/5, K_2 = 2, \gamma_2 = 0)$ MISO BC setting with $L = 2$ antennas. The performance of this setting, as shown in a previous example (Eq. (10)) is $T_2 = 2$.*

This matches the performance of the $L = 2$ antenna homogeneous system with $K = 7$ users and $\gamma = 1/7$, whose delay is again (cf. [3], [4])

$$T_2\left(7, \frac{1}{7}\right) = \frac{K(1 - \gamma)}{L + K\gamma} = 2. \quad (12)$$

B. Coexistence of users with different cache sizes

We now proceed to lift the constraint of cache-less users and consider the more general scenario of $\gamma_2 \in (0, \gamma_1)$.

Theorem 4. *In the L -antenna $(K_1, \gamma_1, K_2, \gamma_2 > 0)$ MISO BC, if $T_{K_1} \geq \frac{K_2(1 - \gamma_2)}{L - 1 + K_2\gamma_2}$, then the achievable delivery time matches that of the corresponding homogeneous system with K users of equally sized caches $\gamma = \frac{K_1\gamma_1 + K_2\gamma_2}{K}$, i.e.,*

$$T_L(K_1, \gamma_1, K_2, \gamma_2) = \frac{K_1(1 - \gamma_1) + K_2(1 - \gamma_2)}{L + K_1\gamma_1 + K_2\gamma_2} \quad (13)$$

while else, the delay takes the form

$$T_L(K_1, \gamma_1, K_2, \gamma_2) = \frac{K_1(1 - \gamma_1)}{K_1\gamma_1 + 1} + \frac{K_2(1 - \gamma_2) - (L - 1 + K_2\gamma_2) \cdot T_{K_1}}{\min\{K_2, L + K_2\gamma_2\}}. \quad (14)$$

Proof. The proof is constructive and is detailed in Section III-B. \square

Remark 1. *Theorems 2 and 4 show how adding either one more antennas or providing caches to the cache-less users, provides the same increase in the DoF. Most importantly, by either increasing the number of antennas or increasing the caches of the weaker users helps to decrease the penalty due to the system heterogeneity and allows to achieve the homogeneous performance. For example, let us assume the L -antenna $(K_1, \gamma_1, L \cdot T_{K_1}, \gamma_2)$ MISO BC where the delay is given by Eq. (7) to be*

$$T_L(K_1, \gamma_1, L \cdot T_{K_1}, 0) = T_{K_1} + \frac{T_{K_1}}{L} = T_{K_1} \frac{L + 1}{L}. \quad (15)$$

We will show that increasing either the number of antennas by 1, or adding a small cache to each of the cache-less users such that $K_2\gamma_2 = 1$, will result in the same DoF performance. First, increasing the number of antennas to $L + 1$, corresponds to the case described by Eq. (8) where

$$T_{L+1}(K_1, \gamma_1, K_2, 0) = T_{K_1} \quad (16)$$

corresponding to a DoF of

$$D_{L+1}(K_1, \gamma_1, L \cdot T_{K_1}, 0) = L + 1 + K_1\gamma_1 \quad (17)$$

Further, in the original setting with L transmit antennas, by adding a small cache to each of the cache-less users such that $K_2\gamma_2 = 1$, we can easily see that the achieved performance corresponds to Eq. (13), thus

$$T_L\left(K_1, \gamma_1, L \cdot T_{K_1}, \frac{1}{L \cdot T_{K_1}}\right) = \frac{K_1(1 - \gamma_1) + K_2 - 1}{L + K_1\gamma_1 + 1}$$

which corresponds to the cache-aided DoF of

$$D_L\left(K_1, \gamma_1, L \cdot T_{K_1}, \frac{1}{L \cdot T_{K_1}}\right) = L + K_1\gamma_1 + 1. \quad (18)$$

From the above remark, we can see that adding antennas or small caches to the cache-less users allows for the full DoF to be achieved. In other words, we can see that the two resources work in tandem to boost the DoF.

Further, the above multiplicative gains can also be achieved in a setting with K_1 users equipped with caches of normalized size γ_1 which coexist with some $K_2 = (\tilde{L} - 1)T_{K_1}$, $\tilde{L} > 1$ cache-less users, by increasing the transmit antennas and/or adding cumulative cache of size Γ_2 to the cache-less users. Specifically, adding these two resources to the system can raise the DoF by a multiplicative factor of $L + \Gamma_2 \leq \tilde{L}$. As we will see, this is a direct outcome of exploiting the multiple antennas as a means of spatially separating users and hence treating *in the same transmission* both user types.

Example 3. Let us assume the single antenna system with $K_2 = 10$ cache-less users and with $K_1 = 7$ cache-aided users equipped with caches of normalized size $\gamma_1 = \frac{1}{7}$. First, we will calculate the performance of the above setting and then we will proceed to add one more antenna, i.e. $L' = 2$ and finally, we will add caches to the cache-less users.

The first setting's performance is calculated (Eq. (5)) to be

$$T_1(7, 1/7, 10, 0) = \frac{7-1}{2} + 10 = 13 \quad (19)$$

while the second setting's performance, given by Eq. (7), is

$$T_2(7, 1/7, 10, 0) = \frac{7-1}{2} + \frac{7}{2} = \frac{13}{2}. \quad (20)$$

Finally, the third setting's performance is given by Eq. (14)

$$T_2(7, 1/7, 10, 1/10) = \frac{7-1}{2} + \frac{3}{3} = 4. \quad (21)$$

From the above we can see that doubling the number of antennas will halve the system delay. Furthermore, if we also equip cache-less users with caches of cumulative size $\Gamma_2 = 1$, while having $L' = 2$ antennas, we can see that the delay is reduced by more than a multiplicative factor of 3, compared to the original setting, which amounts to a multiplicative DoF boost of $L' + \Gamma_2 = 3$ and a further additive reduction attributed to the local caching gain.

III. SCHEME DESCRIPTION

We begin with the scheme description for the case where the cache-aided users co-exist with the cache-less users. This scheme will then serve as the basis for the scheme for the case where both user types have non-trivial cache sizes.

In both of these cases, the challenge of the algorithm lies in properly combining the delivery of content towards each of the two types of users, such that subfiles intended for one set are either "cacheable" or can be "nulled-out" via Zero-Force (ZF) precoding.

Notation: In describing the scheme, we will use the following notation. The file requested by user $k \in [K]$ will be denoted by W^{d_k} . Symbol \oplus denotes the bit-wise XOR operator, \mathbb{N} the set of natural numbers, and for $n, k \in \mathbb{N}$ we denote with $\binom{n}{k}$ the binomial coefficient. For set A we denote its cardinality with $|A|$, while for sets A, B we will use $A \setminus B$ to denote the difference set. We denote with \mathcal{H}_λ^{-1} the normalized inverse ($L \times L$ precoder matrix) of the channel matrix between the L antennas and the L users of some set

$\lambda \subset [K]$, $|\lambda| = L$, where the rows of the precoder matrix are denoted by $\{\mathbf{h}_{\lambda \setminus \{l\}}^\perp\}_{l \in \lambda}$ and have the following property

$$\mathbf{h}_k^H \cdot \mathbf{h}_{\lambda \setminus \{l\}}^\perp = \begin{cases} 1, & \text{if } k = l \\ 0, & \text{if } k \in \lambda \setminus \{l\} \\ \neq 0, & \text{if } k \in [K] \setminus \lambda. \end{cases} \quad (22)$$

We remind that for some $K_i = \{K_1, K_2\}$, we denote by

$$T_{K_i} \triangleq \frac{K_i(1 - \gamma_i)}{1 + K_i\gamma_i}, \quad i \in \{1, 2\} \quad (23)$$

the delay required to treat only one set of users with a single transmit antenna.

Finally for sets $\chi, \beta \subset [K]$, we define XORs X_χ and $X_{\chi, \beta}$ as follows

$$X_\chi = \bigoplus_{k \in \chi} W_{\chi \setminus \{k\}}^{d_k} \quad (24)$$

$$X_{\chi, \beta} = \bigoplus_{k \in \chi} W_{\beta \cup \chi \setminus \{k\}}^{d_k}. \quad (25)$$

A. Placement and delivery in the presence of cache-less users

We will first provide an overview of the algorithm, and will then proceed to describe this algorithm in detail.

1) *Overview of Algorithm:* The first scheme that we will present is designed to serve demands of both cache-aided and cache-less users in the same transmission.

a) *Placement Phase:* The placement phase follows the algorithm in [1] and starts by breaking each file into

$$S = \begin{pmatrix} K_1 \\ K_1\gamma_1 \end{pmatrix} \quad (26)$$

subfiles and then proceeds with storing each subfile at exactly $K_1\gamma_1$ receivers.

b) *Transmission Design:* A transmitted vector is built by first forming an information vector of length L . The first element of this information vector is a XOR, intended for some $K_1\gamma_1 + 1$ cache-aided users, which is designed as in the algorithm of [1] (cf. Eq. (24)), and thus can be decoded by all involved cache-aided users. Further, the remaining messages are $L - 1$ uncoded subfiles, each intended for a different cache-less user. These subfiles are carefully picked to match the file indices associated with the XOR.

Due to the inability of a cache-less user to remove interference, we need to Zero-Force the $L - 1$ unwanted messages (the XOR as well as the $L - 2$ other uncoded messages) to that user. Thus the XOR is Zero-Forced away from the cache-less users, while each uncoded message is Zero-Forced away from $L - 2$ cache-less users and away from one cache-aided user.

As suggested above, thus far we have used-up all the $L - 1$ spatial degrees of freedom for Zero-Forcing the XOR, but we have 1 remaining spatial degree of freedom for each of the uncoded messages. This spatial degree of freedom will be used to ZF all the uncoded messages away from a single cache-aided user. This allows all cache-less users to be able to receive interference-free their intended message, as well as allows one cache-aided user to receive without interference the XORed

message, which it can naturally decode. Most importantly, this last part will allow for the other $K_1\gamma_1$ cache-aided users to be able to have one subfile (per file) in common, which is crucial as we will see.

Since there are no remaining spatial degrees of freedom to exploit, the residual $K_1\gamma_1$ cache-aided users will receive a linear combination of all the L symbols, and will need to exclusively use their cached content to remove these interfering messages. To this end, we need to pick the indices of the uncoded messages in a way such that these messages are completely known to these users. Since a subfile can be cached at a maximum of $K_1\gamma_1$ users and, conversely, a set of $K_1\gamma_1$ cache-aided users can have only one subfile (per file) in common, it follows that the subfile index that is delivered in every cache-less user is the same and is defined exactly by the $K_1\gamma_1$ cache-aided users that we are treating.

c) Verification that all subfiles are transmitted: As we saw in the previous paragraph, each transmission is responsible for communicating one XOR for a set χ of $K_1\gamma_1 + 1$ cache-aided users, as well as $L - 1$ subfiles intended for cache-less users, where these subfiles share the same index τ .

As mentioned before, in order for all $K_1\gamma_1$ users to decode their intended subfile, it suffices to choose the subfile index such that $\tau \subset \chi$. The pairing between a XOR and a subfile index can be viewed as a perfect matching problem over a bipartite graph. In this bipartite graph, a node of the left-hand-side represents one of the XORs (χ), while a node of the right-hand-side represents $L - 1$ subfiles with the same index (τ) that are desired by some $L - 1$ cache-less users.

As we will show later on, this problem is guaranteed to have a solution in our case. While such a solution can be constructed numerically, in the following paragraph we will describe an algorithm that can provide an explicit perfect matching to our problem for any set of parameters. This perfect matching can be achieved by a slight increase of the subpacketization.

2) Algorithm Details: In this section we describe the details of the algorithm, starting from the placement phase. Further, we continue with the delivery algorithm in a pseudo-code format (Alg. 1) accompanied by its description.

a) Placement Phase: Initially, each file W^n , $n \in [N]$, is divided into

$$S_{nc} = K_1(1 - \gamma_1) \binom{K_1}{K_1\gamma_1} \quad (27)$$

subfiles, where these subfiles are named according to

$$W^n \rightarrow \{W_\tau^{n,\phi}, \tau \subset \mathcal{K}_1, |\tau| = K_1\gamma_1, \phi \in \mathcal{K}_1 \setminus \tau\}.$$

Then, cache \mathcal{Z}_k of cache-aided user $k \in \mathcal{K}_1$ is filled according to

$$\mathcal{Z}_k = \{W_\tau^{n,\phi} : k \in \tau, \forall \phi \in \mathcal{K}_1 \setminus \tau, \forall n \in [N]\}. \quad (28)$$

This is identical to the original placement in [1], and the extra subpacketization (corresponding to index ϕ) will facilitate the aforementioned combinatorial problem of matching XORs with uncoded subfiles.

b) Delivery Phase: We will first focus on the case of $K_2 = (L - 1)T_{K_1}$, where the delay

$$T_L(K_1, \gamma_1, K_2, \gamma_2 = 0) = \frac{K_1(1 - \gamma_1) + K_2}{L + K_1\gamma_1} \quad (29)$$

can be achieved by simultaneously treating $K_1\gamma_1 + L$ users. The extension to an arbitrary number of cache-less users is based on the algorithm of the above case, and will be described later on.

Matching Problem: As we have argued, the demands of the cache-aided users are treated by default via each XOR X_χ . At the same time, we are able to treat $L - 1$ cache-less users, under the condition that their received subfile index τ is the same and that $\tau \subset \chi$.

Thus, the challenge presented in the creation of a transmitted vector is to match a XOR index χ with a subfile index τ such that $\tau \subset \chi$ and at the end each χ is matched to a unique τ , in the case where $T_{K_1} = 1$ while if $T_{K_1} > 1$, then χ needs to be matched to one of the T_{K_1} different³ τ . This constitutes a perfect matching over a bipartite graph, where the left-hand-side (LHS) nodes represent the $\binom{K_1}{K_1\gamma_1+1}$ different χ indices, while a node of the right-hand-side (RHS) represents one of the T_{K_1} copies of the $\binom{K_1}{K_1\gamma_1}$ different τ intended for some $L - 1$ cache-less recipients.

This type of problem is guaranteed to have a solution when each node from the LHS is connected to exactly $d \in \mathbb{N}$ nodes of the RHS (see [34]). In our problem, it is easy to see that each node of the LHS is connected to $d = T_{K_1} \cdot (K_1\gamma_1 + 1) = K_1(1 - \gamma_1)$ nodes of the RHS.

Since an algorithm that finds such a solution may have high complexity (for example see [35]) we, instead, present an algorithm that requires a slightly higher subpacketization, but can provide an instant solution to the above matching problem. Specifically, the subpacketization of Eq. (27) contains the term $K_1(1 - \gamma_1)$, thus creating $K_1(1 - \gamma_1)$ copies of each XOR X_χ , while the same holds for every subfile τ intended for the cache-less users. Our algorithm achieves a perfect matching by matching node (ϕ, τ) of the RHS, where $\phi \in \mathcal{K}_1 \setminus \tau$, with one of the XORs $X_{\{\phi\} \cup \tau}$ of the LHS.

c) Transmission: The delivery phase, in the form of pseudo-code, is presented in Alg. 1, which we describe in this paragraph. Transmission commences by splitting the cache-less users into T_{K_1} sets with $L - 1$ users each (Step 2). Then we pick set $\tau \subset \mathcal{K}_1$, $|\tau| = K_1\gamma_1$ (Step 3), where this set serves two purposes. First, it identifies the cache-aided users that will not be assisted by precoding, and second, it identifies the subfile index that the selected cache-less users will receive. Next, cache-aided user ϕ is picked from the remaining set of cache-less users $\mathcal{K}_1 \setminus \tau$ (Step 4). Then, set g_t , containing some $L - 1$ cache-less users, is picked (Step 6).

The transmitted vector is created by calculating the precoder matrix $\mathcal{H}_{\{\phi\} \cup g_t}^{-1}$ such that it forms the normalized inverse of the channel matrix between the L -antenna transmitter and users of set $\{\phi\} \cup g_t$. The precoder matrix is multiplied by the

³We note here that, as discussed before, this index τ is the common index of all subfiles meant for the $L - 1$ cache-less users during this transmission.

Algorithm 1: Transmission in the Cache-less Case

1 Assume $T_{K_1} \in \mathbb{N}$.

2 Group cache-less users:

$$g_1 = \{K_1 + 1, K_1 + 2, \dots, K_1 + L - 1\}, \dots,$$

$$g_{T_{K_1}} = \{K_1 + (L - 1) \cdot (T_{K_1} - 1), \dots, K\}.$$

3 **for all** $\tau \subset \mathcal{K}_1$, $|\tau| = K_1\gamma_1$ (pick file index) **do**

4 **for all** $\phi \in \mathcal{K}_1 \setminus \tau$ (pick precoded user) **do**

5 Set $\chi = \tau \cup \{\phi\}$

6 **for all** $t \in [T_{K_1}]$ (pick cache-less group) **do**

7 Transmit:

$$\mathbf{x}_{\tau, \phi}^t = \mathcal{H}_{\{\phi\} \cup g_t}^{-1} \begin{bmatrix} X_\chi \\ W_\tau^{d_{g_t(1), \phi}} \\ \vdots \\ W_\tau^{d_{g_t(L-1), \phi}} \end{bmatrix}.$$

information vector, which is comprised of XOR X_χ (intended for users $\tau \cup \{\phi\}$) and the $L - 1$ uncoded subfiles that are all indexed by τ (intended for the cache-less users of set g_t) (Step 7).

d) *Decodability:* In each transmitted vector, we can identify two sets of users, i) those that are assisted by precoding (set $\{\phi\} \cup g_t$) and ii) those that are not (set τ). For the “precoding-aided” set, we can immediately recognize that due to the form of the precoder, these users will receive only their intended message. In the special case of the cache-aided user ϕ , decoding the received XOR will also require the use of its cache.

Users belonging to the second set (set τ) will be receiving a linear combination of all L messages i.e.,

$$y_{k \in \tau} = \mathbf{h}_k^H \mathbf{h}_{g_t}^\perp X_\chi + \sum_{i=1}^{L-1} \mathbf{h}_k^H \mathbf{h}_{\lambda \setminus \{g_t(i)\}}^\perp W_\tau^{d_{g_t(i), \phi}} + w_k \quad (30)$$

where $\lambda = \phi \cup g_t$. We can see that all the terms in the summation are cached at all users in set τ , thus can be removed from the equation. What remains is XOR X_χ , which can be decoded (this is direct from [1]) by all members of set χ .

e) *Transmitting unique subfiles every time:* At this point the reader may have noticed that the secondary subfile index, associated with the subfile of the cache-aided users, is not identified in Alg. 1. This is intentional, since every time we transmit subfile $W_\tau^{d_k}$, we pick a new upper index such that all such indices have been picked. We continue to show that the number of times a subfile is transmitted is exactly $K_1 - K_1\gamma_1$.

Proof. Let us assume we are interested in delivering $W_\mu^{d_k}$ to a user belonging to the set of cache-less users. We can see that subfile index μ uniquely defines Step 3, i.e. $\tau = \mu$, while the user’s number, k , uniquely defines Step 6. The algorithm goes over all possible $\phi \in \mathcal{K}_1 \setminus \mu$ (Step 4), thus at the end, different

parts of subfile $W_\mu^{d_k}$ will be delivered exactly $K_1(1 - \gamma_1)$ times to cache-less user k .

Now we turn our focus to some cache-aided user k and examine how many times this user will receive something from subfile $W_\mu^{d_k}$. We need to count the number of times something from this subfile is delivered when user k is assisted by precoding, as well as the number of times parts of this same subfile are transmitted when user k is not assisted by precoding.

When user k is assisted by precoding, it follows that the remaining cache-aided users are uniquely defined by μ , i.e. $\tau = \mu$. Thus, the user’s number defines Step 3 while the user’s request defines Step 4. Then, Algorithm 1 will iterate Step 6 a total of T_{K_1} times.

Further, let us look at the case when user k is not assisted by precoding, which means that $\chi = \{k\} \cup \mu$. As we know, the set of precoded users τ satisfies $\tau \subset \chi$, while there are a total of $K_1\gamma_1 + 1$ different τ for a specific χ . Since user k is not assisted by precoding, it follows that $k \in \tau$, which further means that the number of possible and different τ is $K_1\gamma_1$. For each of these τ , there is a unique ϕ (Step 4) and for each pair (ϕ, τ) Step 6 is iterated a total of T_{K_1} times.

In total, the number of times parts of subfile $W_\mu^{d_k}$ are transmitted – when k is a cache-aided user – is equal to

$$T_{K_1} + K_1\gamma_1 \cdot T_{K_1} = K_1(1 - \gamma_1) \quad (31)$$

which concludes the proof. \square

3) *Scheme generalization and delay calculation:* In this section, we will generalize Alg. 1 to the case of $T_{K_1} \notin \mathbb{N}$ and we will further calculate the delay of the scheme.

a) *Scheme generalization:* We remind that $(L - 1) \cdot T_{K_1}$ represents the threshold beyond which we cannot serve all cache-less users with the maximum DoF. In the case where $(L - 1) \cdot T_{K_1} \notin \mathbb{N}$, it follows that the number of cache-less users K_2 that we can serve using the maximum DoF must be either smaller or higher than $(L - 1) \cdot T_{K_1}$, with both of these cases being treated in the following paragraph. If, on the other hand, $(L - 1) \cdot T_{K_1} \in \mathbb{N}$, while $T_{K_1} \notin \mathbb{N}$, then we can simply increase the subpacketization by a multiplicative factor of $L - 1$. This will create a bipartite graph with $(L - 1)K_1(1 - \gamma_1) \binom{K_1}{K_1\gamma_1 + 1}$ nodes on the LHS and $(L - 1)(K_1\gamma_1 + 1) \binom{K_1}{K_1\gamma_1}$ nodes on the RHS, which means that both numbers are integers, which means that the perfect matching can be achieved.

The other two constraints that we need to address, in order to generalize our algorithm, are the cases where $K_2 \geq (L - 1)T_{K_1}$, also corresponding to the case where $(L - 1) \cdot T_{K_1} \notin \mathbb{N}$.

First, if $K_2 < (L - 1)T_{K_1}$, we proceed as in Alg. 1 but when the demands of the cache-less users have been completely satisfied, then we move to treat only the cache-aided users (through any multi-antenna algorithm, such as [3]–[6]), at a rate of $D_L(K_1, \gamma_1) = L + K_1\gamma_1$ users at a time, thus yielding the overall DoF of $D_L(K_1, \gamma_1, K_2, \gamma_2 = 0) = L + K_1\gamma_1$ for the whole duration of the transmission.

Finally, for the case of $K_2 > (L - 1)T_{K_1}$, delivery is split into two sub-phases. During the first sub-phase, we simply

employ Alg. 1 on the first $(L-1)T_{K_1}$ cache-less users while simultaneously completing the delivery to all K_1 cache-aided users. This is done at a rate of $K_1\gamma_1 + L$ users at a time. Then in the second sub-phase, we treat the remaining $K_2 - (L-1)T_{K_1}$ cache-less users via ZF-precoding, L users at a time. The above sums up to a total delay

$$T_L(K_1, \gamma_1, K_2, \gamma_2 = 0) = T_{K_1} + \frac{K_2 - (L-1)T_{K_1}}{\min\{L, K_2\}}.$$

b) *Delay Calculation:* Following the steps of Alg. 1, corresponding to the case of $K_2 = (L-1)T_{K_1}$, we have

$$\begin{aligned} T_L\left(K_1, \gamma_1, (L-1)\frac{K_1(1-\gamma_1)}{K_1\gamma_1+1}, 0\right) &= \\ &= \frac{\overbrace{\left(\frac{K_1}{K_1\gamma_1}\right)}^{\text{Step 3}} \overbrace{\frac{K_1(1-\gamma_1)}{K_1(1-\gamma_1)}}^{\text{Step 4}} \overbrace{\frac{K_1(1-\gamma_1)}{K_1\gamma_1+1}}^{\text{Step 6}}}{\underbrace{K_1(1-\gamma_1)\left(\frac{K_1}{K_1\gamma_1}\right)}_{\text{Subpacketization}}} \quad (32) \\ &= \frac{K_1(1-\gamma_1)}{K_1\gamma_1+1} = \frac{K_1(1-\gamma_1) + K_2}{K_1\gamma_1 + L}. \quad (33) \end{aligned}$$

B. Scheme description for setting with heterogeneous cache-aided users

In this section we consider the L -antenna MISO BC setting, where both user types are equipped with caches of heterogeneous sizes $\gamma_1, \gamma_2 \in (0, \gamma_1)$. In the context of the single antenna heterogeneous setting, it has been an elusive goal to achieve the performance of the corresponding homogeneous system with $\gamma_{av} = \frac{K_1\gamma_1 + K_2\gamma_2}{K}$ (i.e., of the homogeneous system with the same cumulative cache size constraint). What we will show here is that, for a wide range of parameters, the corresponding performance of the multi-antenna homogeneous setting can indeed be achieved in the multi-antenna heterogeneous setting.

1) *Algorithm overview:* First, we will focus on proving the result of Eq. (13), where we can see that each transmission serves exactly $L + K_1\gamma_1 + K_2\gamma_2$ users.

The main idea is to use the extra spatial degrees of freedom as a way to separate some users that belong to one group from some users that belong to the other group.

As before, we will create an $L \times 1$ information vector, which will be multiplied by an $L \times L$ precoder matrix to form the transmitting vector. The elements of the created vector belong to one of 4 types. One element corresponds to a XOR of $1 + K_1\gamma_1$ subfiles intended for some users of set \mathcal{K}_1 , while another element corresponds to a XOR of $1 + K_2\gamma_2$ subfiles which is intended for some users of set \mathcal{K}_2 . The remaining $L-2$ elements will carry $L_1 - 1$, $L_1 \in [1, L-1]$ uncoded messages for users of set \mathcal{K}_1 and $L_2 - 1$, $L_2 \in [1, L-1]$ uncoded messages for users of set \mathcal{K}_2 , where $L_1 + L_2 = L$, and where the exact values of variables L_1 and L_2 are calculated by solving the equality

$$\frac{K_1(1-\gamma_1)}{L_1 + K_1\gamma_1} = \frac{K_2(1-\gamma_2)}{L_2 + K_2\gamma_2} \quad (34)$$

under the constraint that $L_1 \geq 1$.

In other words, the above solution allocates L_1 streams to the cache-aided users and L_2 streams to the others. This observation will allow us to view the problem at hand as a concatenation of two multi-antenna problems. In what follows, we will make use of a new multi-antenna Coded Caching algorithm corresponding to the homogeneous setting and which we present in detail in Appendix D. Further, we will assume that variables L_1, L_2 are integers, while we relegate the non-integer case to Appendix E.

2) Algorithm Details:

a) *Placement:* We split each file W^n , $n \in [N]$ into

$$S_c = (K_1\gamma_1 + L_1) \binom{K_1}{K_1\gamma_1} (K_2\gamma_2 + L_2) \binom{K_2}{K_2\gamma_2} \quad (35)$$

subfiles, where each subfile $W_{\tau_1, \tau_2}^{n, \phi_1, \phi_2}$ is characterized by 4 indices, $\phi_1 \in [K_1\gamma_1 + L_1]$, $\tau_1 \subset \mathcal{K}_1$, $|\tau_1| = K_1\gamma_1$ and $\phi_2 \in [K_2\gamma_2 + L_2]$, $\tau_2 \subset \mathcal{K}_2$, $|\tau_2| = K_2\gamma_2$, where indices τ_1 and τ_2 reveal which users have cached this subfile from sets \mathcal{K}_1 and \mathcal{K}_2 , respectively, while indices ϕ_1, ϕ_2 will help, as previously in the cache-less case, with the combinatorial problem of matching subfile indices with XORs.

The caches of the users are filled as follows

$$\mathcal{Z}_{k_1 \in \mathcal{K}_1} = \{W_{\tau_1, \tau_2}^{n, \phi_1, \phi_2} : k_1 \in \tau_1, \forall \tau_2, \phi_1, \phi_2\} \quad (36)$$

$$\mathcal{Z}_{k_2 \in \mathcal{K}_2} = \{W_{\tau_1, \tau_2}^{n, \phi_1, \phi_2} : k_2 \in \tau_2, \forall \tau_1, \phi_1, \phi_2\} \quad (37)$$

where it is easy to see that the above placement respects the cache-size constraint of each user.

b) *Delivery Phase:* Algorithm 2 describes the delivery phase in the form of a pseudo-code. We begin by noting that symbol $\beta_{\tau, s}^{(i)} \subseteq [K_i] \setminus \tau$ denotes a set of $L_i - 1$ elements, which are selected to be the elements following $s \in [K_i] \setminus \tau$. For example, assuming that $L_i = 2$ and $[K_i] \setminus \tau = \{1, 2, 3, 4\}$, then $\beta_{\tau, 1}^{(i)} = \{2, 3\}$ and $\beta_{\tau, 3}^{(i)} = \{1, 4\}$. In what follows we will refrain from using the upper index i when describing set β , for the sake of simplicity.

As mentioned before, the algorithm works as a concatenation of two multi-antenna Coded Caching schemes. Specifically, it begins (Step 1) by picking a set of $K_1\gamma_1 + 1$ users ($\chi_1 \subset \mathcal{K}_1$) and then (Step 2) by selecting one of those users ($s_1 \in \chi_1$) to be the precoding-assisted user. The following two steps (Step 3 and Step 4) are responsible for picking a set of $K_2\gamma_2 + 1$ users ($\chi_2 \subset \mathcal{K}_2$) and user $s_2 \in \chi_2$, respectively.

Once these $K_1\gamma_1 + K_2\gamma_2 + 2$ users have been selected, the algorithm proceeds with the calculation of sets $\tau_1 \subset \mathcal{K}_1, \tau_2 \subset \mathcal{K}_2$ which correspond to the set of users that will not be assisted by precoding, as well as proceeds to define the set λ that contains the precoding-assisted users from both sets \mathcal{K}_1 and \mathcal{K}_2 .

In the last step, the algorithm creates the transmitting vector. First, it calculates the normalized inverse of the channel between the L -antenna transmitter and the users of set λ . Then, it forms the information vector which is comprised of L elements. Two of these elements are XORs, X_{χ_1, τ_2} and X_{χ_2, τ_1} , while the remaining are $L_1 - 1$ and $L_2 - 1$

Algorithm 2: Transmission Process for Multi-Antenna Heterogeneous Coded Caching

```

1 for all  $\chi_1 \subseteq [K_1], |\chi_1| = K_1\gamma_1 + 1$  do
2   for all  $s_1 \in \chi_1$  do
3     for all  $\chi_2 \subseteq [K_2], |\chi_2| = K_2\gamma_2 + 1$  do
4       for all  $s_2 \in \chi_2$  do
5         Set:  $\tau_1 = \chi_1 \setminus \{s_1\}$ 
6          $\tau_2 = \chi_2 \setminus \{s_2\}$ 
7          $\lambda = \{s_1\} \cup \{s_2\} \cup \beta_{\tau_1, s_1} \cup \beta_{\tau_2, s_2}$ .
8         Transmit:

```

$$\mathbf{x}_{s_2, \tau_2}^{s_1, \tau_1} = \mathcal{H}_\lambda^{-1} \cdot \begin{bmatrix} X_{\chi_1, \tau_1} \\ W_{\tau_1, \tau_2}^{d_{\beta_{\tau_1, s_1}(1)}} \\ \vdots \\ W_{\tau_1, \tau_2}^{d_{\beta_{\tau_1, s_1}(L_1-1)}} \\ X_{\chi_2, \tau_2} \\ W_{\tau_1, \tau_2}^{d_{\beta_{\tau_2, s_2}(1)}} \\ \vdots \\ W_{\tau_1, \tau_2}^{d_{\beta_{\tau_2, s_2}(L_2-1)}} \end{bmatrix}$$

uncoded messages are respectively intended for some users from set \mathcal{K}_1 and set \mathcal{K}_2 . The transmitting vector is formed as a multiplication of the precoding matrix \mathcal{H}_λ^{-1} with the information vector.

We can see that having selected the XOR for users in \mathcal{K}_1 , along with the precoded user (Steps 1 and 2), then the algorithm goes over all possible combinations of XORs and their respective users corresponding to set \mathcal{K}_2 . In the case of users of set \mathcal{K}_1 , this allows to deliver all the index pairs (ϕ_2, τ_2) that correspond to the other set of users.

c) Decoding Process: The decoding process is similar to that of Alg. 3. For the users in set λ i.e., the precoding-assisted users, we can see that they receive only one of the L messages, thus they either decode using a ZF precoder (users in $\lambda \setminus \{s_1\} \setminus \{s_2\}$) or they use a ZF decoder and continue to decode their respective XOR by use of their cached content (users s_1 and s_2).

The remaining users $(\chi_1 \cup \chi_2 \setminus \{s_1\} \setminus \{s_2\})$ will receive a linear combination of all L information messages, which they can decode using the acquired CSI and their stored content. For example, any user $k \in \tau_1$ will receive

$$y_k = \mathbf{h}_k^H \mathbf{h}_{\lambda \setminus \{s_1\}}^\perp X_{\chi_1, \tau_1} + \mathbf{h}_k^H \sum_{i=1}^{L-1} \mathbf{h}_{\lambda \setminus \beta_{\tau_1, s_1}(i)}^\perp W_{\tau_1, \tau_2}^{d_{\beta_{\tau_1, s_1}(i)}}$$

$$+ \mathbf{h}_k^H \mathbf{h}_{\lambda \setminus \{s_2\}}^\perp X_{\chi_2, \tau_2} + \mathbf{h}_k^H \sum_{i=1}^{L-1} \mathbf{h}_{\lambda \setminus \beta_{\tau_2, s_2}(i)}^\perp W_{\tau_1, \tau_2}^{d_{\beta_{\tau_2, s_2}(i)}} \quad (38)$$

where, excluding the first summand, all the other terms above are completely known to any receiver of set τ_1 , and thus can be removed. The remaining XOR X_{χ_1, τ_2} is decodable by any user in set τ_1 .

C. Extension to the remaining cases

In this section we will prove the result of Eq. (14), which corresponds to the case where the number of streams that should be allocated to the group with the higher cache size is less than one. In this case, we simply treat the users of set \mathcal{K}_1 using one stream, and allocate the remaining $L_2 = L - 1$ streams for the second set of users, \mathcal{K}_2 .

At some point in the transmission, all the files requested by set \mathcal{K}_1 have been successfully communicated, while users of set \mathcal{K}_2 require more transmissions to completely receive their files. This is because we transmit at a rate of $K_1\gamma_1 + 1$ to users of set \mathcal{K}_1 and with rate of $L - 1 + K_2\gamma_2$ to users of set \mathcal{K}_2 , where $\frac{K_1(1-\gamma_1)}{1+K_1\gamma_1} < \frac{K_2(1-\gamma_2)}{L-1+K_2\gamma_2}$.

To complete the transmission of files to the second set of users, we employ any of the multi-antenna cache-aided algorithms serving $L + K_2\gamma_2$ users at a time. Thus, the completion time corresponding to the two sets of transmissions, takes the form

$$T_L(K_1, \gamma_1, K_2, \gamma_2) = \frac{K_1(1-\gamma_1)}{K_1\gamma_1 + 1} + \frac{K_2(1-\gamma_2) - T_1^{(1)}(L-1 + K_2\gamma_2)}{\min\{K_2, L + K_2\gamma_2\}}. \quad (39)$$

IV. EXAMPLES

In this section we will display two examples, one for the cache-less users case, and one for the other case. Both of the examples will present the cases where the full DoF of $D_L(K_1, \gamma_1, K_2, \gamma_2) = L + K_1\gamma_1 + K_2\gamma_2$ is achievable.

We will use the standard notation for user demands, where $A \triangleq W^{d_1}$, $B \triangleq W^{d_2}$, and so on. We will also omit the symbol \oplus in the description of the XORs, so for example, we will write $A_{\tau_1, \tau_2} B_{\tau'_1, \tau'_2}$ instead of $A_{\tau_1, \tau_2} \oplus B_{\tau'_1, \tau'_2}$, etc.

A. Cache-less users example ($\gamma_2 = 0$)

In this example, we will consider the $L = 2$ -antenna MISO BC, where $K_1 = 5$ users have caches of normalized size $\gamma_1 = \frac{1}{5}$, while $K_2 = 2$ users have no caches.

First, each file W^n , $n \in [N]$ is subpacketized into

$$W^n \rightarrow \{W_\tau^{n, \phi}, \tau \subset \mathcal{K}_1, |\tau| = K_1\gamma_1, \phi \in \mathcal{K}_1 \setminus \tau\}. \quad (40)$$

The caches of the users in set \mathcal{K}_1 are filled according to Eq. (28) thus, for example, the cache of the first user contains

$$\mathcal{Z}_1 = \{W_1^{n, 2}, W_1^{n, 3}, W_1^{n, 4}, W_1^{n, 5}, \forall n \in [N]\}.$$

Before we describe the entire sequence of transmitted vectors, we focus briefly on a single vector and its decoding at each user.

a) *Transmission and decoding for a specific set of users:*

The goal is to treat $K_1\gamma_1 + L = 3$ users in each time-slot. Let us look in detail at one transmitted vector, where we treat cache-aided users 1 and 2 together with cache-less user 6. In this case, we transmit

$$\mathbf{x}_{1,2}^1 = \mathcal{H}_{26}^{-1} \begin{bmatrix} A_2^1 B_1^2 \\ F_1^2 \end{bmatrix}. \quad (41)$$

Let us examine the decoding process at the users. First, we can see that User 2 will receive — due to ZF precoding and the design of the precoding matrix \mathcal{H}_{26}^{-1} — only the XORed message $A_2^1 B_1^2$, and can thus proceed to cache-out A_2^1 and decode the desired subfile B_1^2 . User 6 will receive, again due to precoding, only its respective desired message F_1^2 . Finally, User 1 will receive a linear combination of $A_2^1 B_1^2$ and F_1^2 , as follows

$$y_1 = \mathbf{h}_1^H \mathbf{h}_6^\perp A_2^1 B_1^2 + \mathbf{h}_1^H \mathbf{h}_2^\perp F_1^2 + w_1. \quad (42)$$

First, by caching out F_1^2 , User 1 can decode the XOR, and then again by accessing its cache, User 1 can remove B_1^2 to decode its desired message A_2^1 .

b) *Sequence of transmissions:* We now proceed with the entire sequence of the 40 transmissions. Given that each file is subpacketized into $K_1(1 - \gamma_1) \binom{K_1}{K_1\gamma_1} = 5(1 - \frac{1}{5}) \binom{5}{1} = 20$ subpackets, the 40 transmissions will correspond to the desired delay of

$$T_2 \left(5, \frac{1}{5}, 2, 0 \right) = \frac{K_2 + K_1(1 - \gamma_1)}{L + K_1\gamma_1} = 2. \quad (43)$$

The transmissions are:

$$\begin{aligned} \mathbf{x}_{1,2}^1 &= \mathcal{H}_{26}^{-1} \begin{bmatrix} A_2^1 B_1^2 \\ F_1^2 \end{bmatrix}, & \mathbf{x}_{1,2}^2 &= \mathcal{H}_{27}^{-1} \begin{bmatrix} A_2^3 B_1^3 \\ G_1^2 \end{bmatrix} \\ \mathbf{x}_{1,3}^1 &= \mathcal{H}_{36}^{-1} \begin{bmatrix} A_3^1 C_1^3 \\ F_1^3 \end{bmatrix}, & \mathbf{x}_{1,3}^2 &= \mathcal{H}_{37}^{-1} \begin{bmatrix} A_3^2 C_1^2 \\ G_1^3 \end{bmatrix} \\ \mathbf{x}_{1,4}^1 &= \mathcal{H}_{46}^{-1} \begin{bmatrix} A_4^1 D_1^4 \\ F_1^4 \end{bmatrix}, & \mathbf{x}_{1,4}^2 &= \mathcal{H}_{47}^{-1} \begin{bmatrix} A_4^2 D_1^2 \\ G_1^4 \end{bmatrix} \\ \mathbf{x}_{1,5}^1 &= \mathcal{H}_{56}^{-1} \begin{bmatrix} A_5^1 E_1^5 \\ F_1^5 \end{bmatrix}, & \mathbf{x}_{1,5}^2 &= \mathcal{H}_{57}^{-1} \begin{bmatrix} A_5^2 E_1^2 \\ G_1^5 \end{bmatrix} \\ \mathbf{x}_{2,1}^1 &= \mathcal{H}_{16}^{-1} \begin{bmatrix} A_2^4 B_1^4 \\ F_2^1 \end{bmatrix}, & \mathbf{x}_{2,1}^2 &= \mathcal{H}_{17}^{-1} \begin{bmatrix} A_2^5 B_1^5 \\ G_2^1 \end{bmatrix} \\ \mathbf{x}_{2,3}^1 &= \mathcal{H}_{36}^{-1} \begin{bmatrix} B_3^2 C_2^3 \\ F_2^3 \end{bmatrix}, & \mathbf{x}_{2,3}^2 &= \mathcal{H}_{37}^{-1} \begin{bmatrix} B_3^1 C_2^2 \\ G_2^3 \end{bmatrix} \\ \mathbf{x}_{2,4}^1 &= \mathcal{H}_{46}^{-1} \begin{bmatrix} B_4^2 D_2^4 \\ F_2^4 \end{bmatrix}, & \mathbf{x}_{2,4}^2 &= \mathcal{H}_{47}^{-1} \begin{bmatrix} B_4^1 D_2^2 \\ G_2^4 \end{bmatrix} \\ \mathbf{x}_{2,5}^1 &= \mathcal{H}_{56}^{-1} \begin{bmatrix} B_5^2 E_2^5 \\ F_2^5 \end{bmatrix}, & \mathbf{x}_{2,5}^2 &= \mathcal{H}_{57}^{-1} \begin{bmatrix} B_5^1 E_2^2 \\ G_2^5 \end{bmatrix} \\ \mathbf{x}_{3,1}^1 &= \mathcal{H}_{16}^{-1} \begin{bmatrix} A_3^4 C_3^4 \\ F_3^1 \end{bmatrix}, & \mathbf{x}_{3,1}^2 &= \mathcal{H}_{17}^{-1} \begin{bmatrix} A_3^5 C_3^5 \\ G_3^1 \end{bmatrix} \\ \mathbf{x}_{3,2}^1 &= \mathcal{H}_{26}^{-1} \begin{bmatrix} B_3^4 C_2^4 \\ F_3^2 \end{bmatrix}, & \mathbf{x}_{3,2}^2 &= \mathcal{H}_{27}^{-1} \begin{bmatrix} B_3^5 C_2^5 \\ G_3^2 \end{bmatrix} \\ \mathbf{x}_{3,4}^1 &= \mathcal{H}_{46}^{-1} \begin{bmatrix} C_4^3 D_3^4 \\ F_3^4 \end{bmatrix}, & \mathbf{x}_{3,4}^2 &= \mathcal{H}_{47}^{-1} \begin{bmatrix} C_4^1 D_3^2 \\ G_3^4 \end{bmatrix} \\ \mathbf{x}_{3,5}^1 &= \mathcal{H}_{56}^{-1} \begin{bmatrix} C_5^3 E_3^5 \\ F_3^5 \end{bmatrix}, & \mathbf{x}_{3,5}^2 &= \mathcal{H}_{57}^{-1} \begin{bmatrix} C_5^1 E_3^2 \\ G_3^5 \end{bmatrix} \end{aligned}$$

$$\begin{aligned} \mathbf{x}_{4,1}^1 &= \mathcal{H}_{16}^{-1} \begin{bmatrix} A_4^3 D_1^3 \\ F_4^1 \end{bmatrix}, & \mathbf{x}_{4,1}^2 &= \mathcal{H}_{17}^{-1} \begin{bmatrix} A_4^5 D_1^5 \\ G_4^1 \end{bmatrix} \\ \mathbf{x}_{4,2}^1 &= \mathcal{H}_{26}^{-1} \begin{bmatrix} B_4^3 D_2^3 \\ F_4^2 \end{bmatrix}, & \mathbf{x}_{4,2}^2 &= \mathcal{H}_{27}^{-1} \begin{bmatrix} B_4^5 D_2^5 \\ G_4^2 \end{bmatrix} \\ \mathbf{x}_{4,3}^1 &= \mathcal{H}_{36}^{-1} \begin{bmatrix} C_4^2 D_3^2 \\ F_4^3 \end{bmatrix}, & \mathbf{x}_{4,3}^2 &= \mathcal{H}_{37}^{-1} \begin{bmatrix} C_4^5 D_3^5 \\ G_4^3 \end{bmatrix} \\ \mathbf{x}_{4,5}^1 &= \mathcal{H}_{56}^{-1} \begin{bmatrix} D_5^4 E_4^5 \\ F_4^5 \end{bmatrix}, & \mathbf{x}_{4,5}^2 &= \mathcal{H}_{57}^{-1} \begin{bmatrix} D_5^1 E_4^1 \\ G_4^5 \end{bmatrix} \\ \mathbf{x}_{5,1}^1 &= \mathcal{H}_{16}^{-1} \begin{bmatrix} A_5^3 E_1^3 \\ F_5^1 \end{bmatrix}, & \mathbf{x}_{5,1}^2 &= \mathcal{H}_{17}^{-1} \begin{bmatrix} A_5^4 E_1^4 \\ G_5^1 \end{bmatrix} \\ \mathbf{x}_{5,2}^1 &= \mathcal{H}_{26}^{-1} \begin{bmatrix} B_5^3 E_2^3 \\ F_5^2 \end{bmatrix}, & \mathbf{x}_{5,2}^2 &= \mathcal{H}_{27}^{-1} \begin{bmatrix} B_5^5 E_2^5 \\ G_5^2 \end{bmatrix} \\ \mathbf{x}_{5,3}^1 &= \mathcal{H}_{36}^{-1} \begin{bmatrix} C_5^2 E_3^2 \\ F_5^3 \end{bmatrix}, & \mathbf{x}_{5,3}^2 &= \mathcal{H}_{37}^{-1} \begin{bmatrix} C_5^4 E_3^4 \\ G_5^3 \end{bmatrix} \\ \mathbf{x}_{5,4}^1 &= \mathcal{H}_{46}^{-1} \begin{bmatrix} D_5^2 E_4^2 \\ F_5^4 \end{bmatrix}, & \mathbf{x}_{5,4}^2 &= \mathcal{H}_{47}^{-1} \begin{bmatrix} D_5^3 E_4^3 \\ G_5^4 \end{bmatrix}. \end{aligned}$$

The 40 slots, each of normalized duration

$$t_s = \left[(K_1(1 - \gamma_1) \binom{K_1}{K_1\gamma_1}) \right]^{-1} = \frac{1}{20}$$

imply a delay $T_2(5, \frac{1}{5}, 2, 0) = 2$, which matches the delay

$$T_2 \left(7, \frac{1}{7} \right) = \frac{7-1}{2+1} = \frac{K(1 - \gamma_{av})}{2 + K\gamma_{av}} = 2$$

that would be needed in the homogeneous case where the $K = 7$ users would have an identical $\gamma_{av} = \frac{1}{7}$ (same cumulative cache $K\gamma_{av} = 1$).

B. Two Type Cache-aided Example

In this section we present an example that illustrates the mechanics of the two user case. Specifically, we will focus on the $L = 3$ -antenna MISO BC, where $K_1 = 5$ users of set \mathcal{K}_1 are equipped with caches of normalized size $\gamma_1 = \frac{2}{5}$, while $K_2 = 4$ users of set \mathcal{K}_2 are equipped with caches of normalized size $\gamma_2 = \frac{1}{4}$. For this setting, the number of streams (cf. Eq. (34)) should be divided as $L_1 = 1$ and $L_2 = 2$.

We begin by splitting each file into

$$S = (K_1\gamma_1 + L_1)(K_1\gamma_2 + L_2) \binom{K_1}{K_1\gamma_1} \binom{K_2}{K_2\gamma_2} = 360$$

subfiles, where subfile $W_{\tau_1, \tau_2}^{n, \phi_1, \phi_2}$ has indices $\tau_1 \subset [5]$, $|\tau_1| = 2$, $\phi_1 \in [3]$, $\tau_2 \subset [4]$, $|\tau_2| = 1$, $\phi_2 \in [3]$.

1) *Placement Phase:* This phase is carried out according to Eq. (36)-(37) where, for example, the caches of users $1 \in \mathcal{K}_1$ and $6 \in \mathcal{K}_2$ are filled as

$$\begin{aligned} \mathcal{Z}_1 &= \{W_{12, \tau_2}^{n, \phi_1, \phi_2}, W_{13, \tau_2}^{n, \phi_1, \phi_2}, W_{14, \tau_2}^{n, \phi_1, \phi_2}, W_{15, \tau_2}^{n, \phi_1, \phi_2}, \forall \tau_2, \phi_1, \phi_2\} \\ \mathcal{Z}_6 &= \{W_{\tau_1, 6}^{n, \phi_1, \phi_2}, \forall \tau_1, \phi_1, \phi_2\}. \end{aligned}$$

2) *Delivery Phase:* For notational simplicity, we abstain from using indices ϕ_1, ϕ_2 . Further, we will only present one iteration of the algorithmic steps 1-2, that delivers the first

XOR ($A_{23}B_{13}C_{12}$ intended for users 1, 2, 3) of the user set \mathcal{K}_1 , while it goes through all other steps (Steps 3-8).

$$\begin{aligned} \mathbf{x}_{1,23}^{6,7} &= \mathcal{H}_{168}^{-1} \begin{bmatrix} A_{23,7}B_{13,7}C_{12,7} \\ F_{23,7}G_{23,6} \\ H_{23,7} \end{bmatrix} \\ \mathbf{x}_{1,23}^{7,6} &= \mathcal{H}_{178}^{-1} \begin{bmatrix} A_{23,6}B_{13,6}C_{12,6} \\ F_{23,7}G_{23,6} \\ H_{23,6} \end{bmatrix} \\ \mathbf{x}_{1,23}^{6,8} &= \mathcal{H}_{167}^{-1} \begin{bmatrix} A_{23,8}B_{13,8}C_{12,8} \\ F_{23,8}H_{23,6} \\ G_{23,8} \end{bmatrix} \\ \mathbf{x}_{1,23}^{8,6} &= \mathcal{H}_{187}^{-1} \begin{bmatrix} A_{23,6}B_{13,6}C_{12,6} \\ F_{23,8}H_{23,6} \\ G_{23,6} \end{bmatrix} \\ \mathbf{x}_{1,23}^{6,9} &= \mathcal{H}_{167}^{-1} \begin{bmatrix} A_{23,9}B_{13,9}C_{12,9} \\ F_{23,9}I_{23,6} \\ G_{23,9} \end{bmatrix} \\ \mathbf{x}_{1,23}^{9,6} &= \mathcal{H}_{197}^{-1} \begin{bmatrix} A_{23,6}B_{13,6}C_{12,6} \\ F_{23,9}I_{23,6} \\ G_{23,6} \end{bmatrix} \\ \mathbf{x}_{1,23}^{7,8} &= \mathcal{H}_{179}^{-1} \begin{bmatrix} A_{23,8}B_{13,8}C_{12,8} \\ G_{23,8}H_{23,7} \\ I_{23,8} \end{bmatrix} \\ \mathbf{x}_{1,23}^{8,7} &= \mathcal{H}_{189}^{-1} \begin{bmatrix} A_{23,7}B_{13,7}C_{12,7} \\ G_{23,8}H_{23,7} \\ I_{23,7} \end{bmatrix} \\ \mathbf{x}_{1,23}^{7,9} &= \mathcal{H}_{178}^{-1} \begin{bmatrix} A_{23,9}B_{13,9}C_{12,9} \\ G_{23,9}I_{23,7} \\ H_{23,9} \end{bmatrix} \\ \mathbf{x}_{1,23}^{9,7} &= \mathcal{H}_{196}^{-1} \begin{bmatrix} A_{23,7}B_{13,7}C_{12,7} \\ G_{23,9}I_{23,7} \\ F_{23,7} \end{bmatrix} \\ \mathbf{x}_{1,23}^{8,9} &= \mathcal{H}_{186}^{-1} \begin{bmatrix} A_{23,9}B_{13,9}C_{12,9} \\ H_{23,9}I_{23,8} \\ F_{23,9} \end{bmatrix} \\ \mathbf{x}_{1,23}^{9,8} &= \mathcal{H}_{196}^{-1} \begin{bmatrix} A_{23,8}B_{13,8}C_{12,8} \\ H_{23,9}I_{23,8} \\ F_{23,8} \end{bmatrix}. \end{aligned}$$

C. Decoding Process

The decoding process follows the decoding steps of Alg. 3. First, the members of set λ , i.e. the precoding-assisted users, will receive one of the L messages, which can decode using their cached content.

Further, the users of set $\tau_1 \cup \tau_2$ will receive a linear combination of all L messages, which can decode using the acquired CSIT and their cached content.

As an example, we will look at the decoding of transmitted message $\mathbf{x}_{1,23}^{6,7}$ at any intended user. First, we can see that the

precoded users are 1, 6, 8 and these users will receive

$$y_{k \in \{1,6,8\}} = \mathbf{h}_k^H \mathbf{h}_{\{1,6,8\} \setminus \{k\}}^\perp \begin{cases} A_{23,7}B_{13,7}C_{12,7}, & k = 1 \\ F_{23,7}G_{23,6}, & k = 6 \\ H_{23,7}, & k = 8 \end{cases}$$

where naturally any of these users can decode its intended subfile.

Then, for the remaining users (users 2, 3, 7) the received signal takes the form

$$y_{k \in \{2,3,7\}} = \mathbf{h}_k^H \mathbf{h}_{\{6,8\}}^\perp A_{23,7}B_{13,7}C_{12,7} + \mathbf{h}_k^H \mathbf{h}_{\{1,8\}}^\perp F_{23,7}G_{23,6} + \mathbf{h}_k^H \mathbf{h}_{\{1,6\}}^\perp H_{23,7} + w_k. \quad (44)$$

We can easily see that each of these users can decode its desired subfile by caching-out any other interfering message.

V. CONCLUSION AND FINAL REMARKS

An interesting outcome (Theorem 3) is the fact that despite having abundant side information at a sizable number K_1 of receivers, going from 1 to L antennas gives an L -fold DoF boost. This comes in obvious contrast to the cache-aided multiple antenna setting with only cache-aided users [3]–[6], where adding antennas increases additively and not multiplicatively the DoF.

Furthermore, we showed that adding antennas can ameliorate and even remove the effects of cache-size asymmetries. This can be important in practical scenarios where γ_1 is expected to be small, which would then allow cache-aided users to boost the DoF performance of a large number ($\approx (L-1)/\gamma_1$) of cache-less users. Finally we have seen in Theorem 4 the powerful (multiplicative) DoF effect that modest amounts of caching can have in uneven cache-size scenarios.

Remark 2. *The delay of the single-antenna system (K_1, γ_1) BC is the same as the L -antenna MISO BC with extra added $K_2 = (L-1)T_{K_1} \approx \frac{L-1}{\gamma_1}$ users.*

The above remark says that for every new antenna we add to the system, we can also treat an additional, fixed number of approximately $\frac{1}{\gamma_1}$ cache-less users *without* increasing the overall delay.

Intuition on the Cache-less user Design

The algorithm, which is either optimal or near optimal, manages to achieve full coding gains by eliminating the previously encountered penalties of cache-size unevenness. Key to this, was the careful use of antenna-aided user separation.

In the case where cache-aided and cache-less users coexist, the scheme employs this separation in two ways. First, the scheme protects the cache-less users from the XOR and from each other. Secondly, and most importantly, separation allowed $K_1\gamma_1$ cache-aided users to be able to have one subfile per file in common. Even though the employed cache-aided users can cache-out interfering subfiles, nevertheless any collection of $K_1\gamma_1 + 1$ of such users do not have any common subfile index cached, since a subfile is cached at exactly $K_1\gamma_1$ users.

This obstacle was surmounted by Zero-Forcing the messages intended for the cache-less users away from one cache-aided user. This allowed for the aforementioned ability for the cache-aided users (that are not protected by precoding) to share a common subfile index, and thus, by design, to cache out all the subfiles intended for the cache-less users.

APPENDIX A PROOF OF THEOREM 1

Toward proving Theorem 1, we adapt the approach of [31], to lower bound the delay for the case where, out of the K users, only K_1 users have a cache. The bound will then also prove tight for all

$$L \geq \frac{K_2}{T_{K_1}} - 1 = \frac{K_2(1 + K_1\gamma)}{K_1(1 - \gamma)} - 1 \quad (45)$$

as we will see in Appendix C.

The proof (for $L = 1$) tracks closely steps⁴ from [31] which — for the case of $K_1 = K$ (where all users have caches) — employed index coding to bound the performance of coded caching. Some of these steps are sketched here for the sake of completeness. Particular care is taken here to properly construct the bound's counting arguments in a way that accounts for the fact that specific symmetries that are essential to the approach in [31], do not directly hold here, simply because the set $\mathcal{K}_1 = [K_1]$ of users that enjoy side information is only a subset of the users that request files.

We will begin with lower bounding, first for the case of $L = 1$, the delay $T(\mathbf{d}, \chi)$ for any generic caching-delivery strategy χ and any demand vector $\mathbf{d} \in \mathcal{D}_{wc} \triangleq \{\mathbf{d} : d_i \neq d_j, i, j \in [K], i \neq j\}$ whose K entries are all different. In the following, we use \mathcal{Z}_i to denote the cache of each user i , where naturally $\mathcal{Z}_i = \emptyset$ for $i \in \mathcal{K}_2 \triangleq [K] \setminus \mathcal{K}_1$.

Distinct caching problems and their corresponding index coding equivalents : We first follow closely the approach in [31] to describe the association between index coding and our specific caching scenario here. As in [31], each caching problem (defined by a demand vector $\mathbf{d} \in \mathcal{D}_{wc}$) is converted into an index coding problem, by having each requested file W^{d_i} split into 2^{K_1} disjoint subfiles $W_{\mathcal{T}}^{d_i}, \mathcal{T} \in 2^{[K_1]}$, where $\mathcal{T} \subseteq [K_1]$ indicates the set of users that have $W_{\mathcal{T}}^{d_i}$ cached. Since no subfile of the form $W_{\mathcal{T}}^{d_i}, \mathcal{T} \ni i$ is requested, the index coding problem here is defined by

$$K_1 2^{K_1-1} + K_2 2^{K_1}$$

requested subfiles, which form the nodes of the side-information graph $\mathcal{G} = (\mathcal{V}_{\mathcal{G}}, \mathcal{E}_{\mathcal{G}})$, where $\mathcal{V}_{\mathcal{G}}$ is the set of vertices (each vertex/node representing a different demanded subfile $W_{\mathcal{T}}^{d_i}, \mathcal{T} \not\ni i$) and $\mathcal{E}_{\mathcal{G}}$ is the set of direct edges of the

⁴We note in advance that a naive adaptation of the approach in [31], where we would simply account for a reduced sum-cache constraint $K_1 M$ corresponding to a redundancy $t = \frac{K_1 M}{N}$, would yield a loose bound; for example when $L = 1$, this naive bound would be $T \geq \frac{K-t}{t+1}$ which would then translate to $T \geq \frac{K_1(1-\gamma)}{1+K_1\gamma} + \frac{K_2}{1+K_1\gamma}$ which is loose as the optimal delay will turn out to be $T = \frac{K_1(1-\gamma)}{1+K_1\gamma} + K_2$.

graph. We recall that an edge from node $W_{\mathcal{T}}^{d_i}$ to $W_{\mathcal{T}'}^{d_{i'}}$ exists if and only if $i' \in \mathcal{T}$.

As in [31], this allows us to lower bound $T(\mathbf{d}, \chi)$ by using the index-coding converse from [36] which says that for a given \mathbf{d}, χ — with corresponding side information graph $\mathcal{G}_{\mathbf{d}} = (\mathcal{V}_{\mathcal{G}}, \mathcal{E}_{\mathcal{G}})$ with $\mathcal{V}_{\mathcal{G}}$ vertices/nodes and $\mathcal{E}_{\mathcal{G}}$ edges — the delay is bounded as

$$T \geq \sum_{v \in \mathcal{V}_{\mathcal{G}}} |v| \quad (46)$$

for every acyclic induced subgraph \mathcal{J} of $\mathcal{G}_{\mathbf{d}}$, where $\mathcal{V}_{\mathcal{J}}$ denotes the set of nodes of the subgraph \mathcal{J} , and where $|v|$ is the size of the message/subfile/node v .

The following describes the acyclic graphs, and also directly shows that these remain acyclic after they are enlarged to account for the content requested by the cache-less users. In the following we will consider permutations $\sigma \in S_{K_1}$ from the symmetric group S_{K_1} , and, for a given demand vector \mathbf{d} , we will use $\mathcal{A}_{\mathbf{d}} \triangleq \cup_{i \in [K] \setminus [K_1]} W^{d_i}$ to denote the union of all content in \mathbf{d} that is requested by the users in $[K] \setminus [K_1]$.

Lemma 1. *For any \mathbf{d} and any $\sigma \in S_{K_1}$, and for an acyclic subgraph $\mathcal{J}_{\mathbf{d}, \sigma}$ of $\mathcal{G}_{\mathbf{d}}$, is designed here to consist of all subfiles $\{W_{\mathcal{T}}^{d_{\sigma(i)}}, \forall i \in [K_1], \forall \mathcal{T} \subseteq [K_1] \setminus \{\sigma(1), \sigma(2), \dots, \sigma(i)\}\}$, then the enlarged graph $\mathcal{J}_{\mathbf{d}, \sigma} \cup \mathcal{A}_{\mathbf{d}}$ is also acyclic.*

Proof. The proof that the subgraph $\mathcal{J}_{\mathbf{d}, \sigma}$ is acyclic is direct from [31, Lemma 1]. The proof that $\mathcal{J}_{\mathbf{d}, \sigma} \cup \mathcal{A}_{\mathbf{d}}$ is also an acyclic graph, i.e., that the addition (on the original $\mathcal{J}_{\mathbf{d}, \sigma}$) of all the nodes corresponding to $\mathcal{A}_{\mathbf{d}}$ does not induce any cycles, follows by first recalling that a directed edge from node $W_{\mathcal{T}}^{d_i}$ to $W_{\mathcal{T}'}^{d_{i'}}$ exists if and only if $i' \in \mathcal{T}$, which thus tells us that an edge cannot be drawn from any node representing content from $\mathcal{A}_{\mathbf{d}}$, because any cache-less user $i \in [K] \setminus [K_1]$ cannot belong to any such \mathcal{T} simply because $\mathcal{T} \subseteq [K_1]$. \square

Given the acyclic subgraph $\mathcal{J}_{\mathbf{d}, \sigma} \cup \mathcal{A}_{\mathbf{d}}$, we combine Lemma 1 with (46) to get

$$T(\mathbf{d}, \chi) \geq T^{LB}(\sigma, \mathbf{d}, \chi) \quad (47)$$

where

$$\begin{aligned} T^{LB}(\sigma, \mathbf{d}, \chi) &\triangleq \sum_{v \in \mathcal{V}_{\mathcal{J}_{\mathbf{d}, \sigma} \cup \mathcal{A}_{\mathbf{d}}}} |v| \\ &= \sum_{\mathcal{T} \subseteq [K_1] \setminus \{\sigma(1)\}} |W_{\mathcal{T}}^{d_{\sigma(1)}}| + \sum_{\mathcal{T} \subseteq [K_1] \setminus \{\sigma(1), \sigma(2)\}} |W_{\mathcal{T}}^{d_{\sigma(2)}}| + \dots \\ &+ \sum_{\mathcal{T} \subseteq [K_1] \setminus \{\sigma(1), \dots, \sigma(K_1)\}} |W_{\mathcal{T}}^{d_{\sigma(K_1)}}| + |\mathcal{A}_{\mathbf{d}}|. \end{aligned} \quad (48)$$

Then, as in [31], we average over worst-case demands to get

$$\begin{aligned} T^* &\triangleq \min_{\chi} \max_{\mathbf{d} \in \mathcal{D}_{wc}} T(\mathbf{d}, \chi) \\ &\geq \min_{\chi} \max_{\mathbf{d} \in \mathcal{D}_{wc}} \max_{\sigma \in S_{K_1}} T^{LB}(\sigma, \mathbf{d}, \chi) \\ &\geq \min_{\chi} \frac{1}{|\mathcal{D}_{wc}|} \frac{1}{|S_{K_1}|} \sum_{\sigma \in S_{K_1}} \sum_{\mathbf{d} \in \mathcal{D}_{wc}} T^{LB}(\sigma, \mathbf{d}, \chi) \end{aligned}$$

$$\geq \min_{\chi} \frac{1}{P(N, K)K_1!} \sum_{\sigma \in S_{K_1}} \sum_{\mathbf{d} \in \mathcal{D}_{W_c}} T^{LB}(\sigma, \mathbf{d}, \chi) \quad (49)$$

where in the above we use $P(N, K) \triangleq \frac{N!}{(N-K)!}$.

Rewriting the summation in (49), we get

$$\begin{aligned} \sum_{\sigma \in S_{K_1}} \sum_{\mathbf{d} \in \mathcal{D}_{W_c}} T^{LB}(\sigma, \mathbf{d}, \chi) &= \\ \sum_{i=0}^{K_1} \sum_{n \in [N]} \sum_{\mathcal{T} \subseteq [K_1]: |\mathcal{T}|=i} |W_{\mathcal{T}}^n| \cdot \underbrace{\sum_{\sigma \in S_{K_1}} \sum_{\mathbf{d} \in \mathcal{D}_{W_c}} \mathbb{1}_{\mathcal{V}_{\mathcal{J}_{\mathbf{d}, \sigma}}}(W_{\mathcal{T}}^n) + |\mathcal{A}_{\mathbf{d}}|}_{\triangleq Q_i(W_{\mathcal{T}}^n)} \end{aligned} \quad (50)$$

where $\mathcal{V}_{\mathcal{J}_{\mathbf{d}, \sigma}}$ is the set of vertices in the acyclic component subgraph $\mathcal{J}_{\mathbf{d}, \sigma}$ for a given \mathbf{d}, σ pair, and where $\mathbb{1}_{\mathcal{V}_{\mathcal{J}_{\mathbf{d}, \sigma}}}(W_{\mathcal{T}}^n)$ denotes the indicator function which takes the value of 1 only if $W_{\mathcal{T}}^n \subset \mathcal{V}_{\mathcal{J}_{\mathbf{d}, \sigma}}$, else it is set to zero.

Counting arguments accounting for cache-less users :

Our aim is to count the number of times, $Q_i(W_{\mathcal{T}}^n)$, that any specific subfile $W_{\mathcal{T}}^n$ appears in the summation in (50). To do this, we draw from the counting arguments in [37, Section VII-C] which derives $Q_i(W_{\mathcal{T}}^n)$ for the case where K users share $\Lambda \leq K$ caches, where each cache $r \subset [\Lambda]$ serves Λ_r users. Adapting these steps⁵ in [37, Section VII-C] gives that

$$\begin{aligned} Q_i &= Q_i(W_{\mathcal{T}}^n) \triangleq \sum_{\sigma \in S_{K_1}} \sum_{\mathbf{d} \in \mathcal{D}_{W_c}} \mathbb{1}_{\mathcal{V}_{\mathcal{J}_{\mathbf{d}, \sigma}}}(W_{\mathcal{T}}^n) \\ &= \binom{N-1}{K-1} \sum_{r=1}^{K_1} P(K_1 - i - 1, r - 1)(K_1 - r)! \\ &\quad \cdot (K-1)!(K_1-1)!(K_1-i). \end{aligned} \quad (51)$$

Setting $x_i \triangleq \sum_{n \in [N]} \sum_{\mathcal{T} \subseteq [K_1]: |\mathcal{T}|=i} |W_{\mathcal{T}}^n|$ and recalling that

$$N = \sum_{i=0}^{K_1} x_i = \sum_{i=0}^{K_1} \sum_{n \in [N]} \sum_{\mathcal{T} \subseteq [K_1]: |\mathcal{T}|=i} |W_{\mathcal{T}}^n| \quad (52)$$

we combine (49), (50) and (51), to get

$$T \geq \sum_{i=0}^{K_1} \frac{Q_i}{P(N, K)K_1!} x_i. \quad (53)$$

We now resume counting to calculate $\frac{Q_i}{\Lambda!P(N, K)}$ for each $i = 0, 1, \dots, K_1$. Applying (51), we see that

$$\begin{aligned} \frac{Q_i}{K_1!P(N, K)} &= \frac{(N-1)!(N-K)!}{(K-1)!(N-K)!K_1!N!} \\ &\quad \cdot \sum_{r=1}^{K_1} (K-1)!(K_1-i)P(K_1-i-1, r-1)(K_1-r)! \\ &= \frac{1}{K_1!N} \sum_{r=1}^{K_1} (K_1-i)P(K_1-i-1, r-1)(K_1-r)! \end{aligned}$$

⁵The following expression could not have been derived, had we simply substituted K for K_1 , in the corresponding Q_i expression in [31]. Such a naive approach would have essentially corresponded to treating the cache-less and cache-aided cases separately, and would not have allowed us to guarantee, among other things, that both cache-less and cache-aided users request different files.

$$\begin{aligned} &= \frac{1}{K_1!N} \sum_{r=1}^{K_1} \frac{(K_1-i)(K_1-i-1)!(K_1-r)!}{(K_1-i-r)!} \\ &= \frac{1}{K_1!N} \sum_{r=1}^{K_1} \frac{(K_1-i)!(K_1-r)!}{(K_1-i-r)!} \\ &= \frac{1}{N} \sum_{r=1}^{K_1} \frac{(K_1-i)!(K_1-r)!}{K_1!(K_1-i-r)!} \\ &= \frac{1}{N} \sum_{r=1}^{K_1} \frac{\binom{K_1-r}{i}}{\binom{K_1}{i}} = \frac{\binom{K_1}{i+1}}{\binom{K_1}{i}N} = \frac{K_1-i}{(i+1)N}. \end{aligned} \quad (54)$$

Now substituting (54) into (53), we get that

$$T(\chi) \geq \sum_{i=0}^{K_1} \frac{K_1-i}{(i+1)N} x_i + \frac{K_1!P(N, K)}{K_1!P(N, K)} \underbrace{|\mathcal{A}_{\mathbf{d}}|}_{K_2} \quad (55)$$

where the use of the fraction $\frac{K_1!P(N, K)}{K_1!P(N, K)} = 1$ is meant to remind us the number of times acyclic graphs corresponding to $\mathcal{A}_{\mathbf{d}}$ were invoked in the summation in (50), and where we also note that the expression above follows from the fact that all $\mathbf{d} \in \mathcal{D}_{W_c}$ force $|\mathcal{A}_{\mathbf{d}}| = K_2$.

Optimization: At this point we observe that the crucial constant $\frac{K_1-i}{(i+1)N}$ derived for the part of the subgraph corresponding to cache-aided users, matches exactly the number $\frac{K_1-i}{(i+1)N}$ derived in [31] for the $K = K_1$ case where all users can have a cache. Consequently, under the same file-size constraint given in (52), and given the current cache-size constraint $\sum_{i=0}^{K_1} i \cdot x_i \leq K_1M$, the expression in (55) serves as a lower bound on the delay of scheme χ whose cache placement implies the set $\{x_i\}$.

Then, following the exact minimization steps in [32, Proof of Lemma 2], we get

$$T(\chi) \geq \frac{K_1(1-\gamma)}{1+K_1\gamma} + K_2 \quad (56)$$

for integer $K_1\gamma$, whereas for all other values of $K_1\gamma$, this is extended to its convex lower envelop.

This concludes lower bounding $\max_{\mathbf{d} \in \mathcal{D}_{W_c}} T(\mathbf{d}, \chi)$, and thus — given that the right hand side of (56) is independent of χ — lower bounds the performance for any scheme χ , hence concluding the proof of the converse for Theorem 1 for the case of $L = 1$.

APPENDIX B

CONVERSE AND GAP TO OPTIMAL FOR THEOREM 2

Let us first consider the gap to optimal for the case of $K_2 \geq (L-1)T_{K_1}$.

We have seen that when $K_2 = \alpha(L-1)T_{K_1}$, $\alpha \geq 1$, the achievable delay in (7) takes the form

$$T_L(K_1, \gamma_1, K_2, \gamma_2) = T_{K_1} + \frac{K_2 - (L-1)T_{K_1}}{L} \quad (57)$$

$$= T_{K_1} + (\alpha - 1) \frac{L-1}{L} T_{K_1} \quad (58)$$

$$= \frac{T_{K_1}}{L} (\alpha L - \alpha + 1). \quad (59)$$

For a lower bound on the minimum possible delay, we use

$$T^* = \frac{\min\{K_2, L\}}{L} = \min \left\{ 1, \frac{\alpha(L-1)T_{K_1}}{L} \right\} \quad (60)$$

corresponding to the optimal delay required to satisfy only the cache-less users. A quick calculation of the ratio between (57) and (60), bounds the gap as

$$G = \frac{\alpha L - \alpha + 1}{\alpha L - \alpha} = 1 + \frac{1}{\alpha(L-1)} \leq 2. \quad (61)$$

When $K_2 < L$, then $\alpha(L-1)T_{K_1} < L$, which again gives

$$G = \frac{T_{K_1}(\alpha L - \alpha + 1)}{L} < \frac{T_{K_1}(\alpha L - \alpha - 1)}{\alpha(L-1)T_{K_1}} \leq 2. \quad (62)$$

For the case of $K_2 = \alpha(L-1)T_{K_1}$, $\alpha \leq 1$, the lower bound takes the form

$$T_L(K_1, \gamma_1, K_2, \gamma_2) \geq \max \left\{ \frac{\min\{L, K_2\}}{L}, \frac{1}{2} \frac{K_1(1-\gamma_1)}{K_1\gamma_1 + L} \right\} \quad (63)$$

where the first term corresponds to the optimal performance of an ‘easier’ system where all the cache-aided users are removed, and where the second term corresponds to an easier system where all cache-less users are removed, and where — for this latter type of system — we know from [4] that treating $K_1\gamma_1 + L$ users at a time is at most a factor of 2 from optimal, under the assumptions of linear and one-shot schemes. Combining (63) with the achievable

$$T_L(K_1, \gamma_1, K_2, \gamma_2) = \frac{K_1 + K_1(1-\gamma_1)}{K_1\gamma_1 + L} \quad (64)$$

from (8), yields a gap of

$$G = \frac{\frac{K_2 + K_1(1-\gamma_1)}{K_1\gamma_1 + L}}{\max \left\{ \frac{K_2}{L}, \frac{1}{2} \frac{K_1(1-\gamma_1)}{K_1\gamma_1 + L} \right\}}.$$

To bound this gap, note that if $\frac{K_2}{L} > \frac{1}{2} \frac{K_1(1-\gamma_1)}{K_1\gamma_1 + L}$ then we know from before that

$$G = \frac{\frac{K_2 + K_1(1-\gamma_1)}{K_1\gamma_1 + L}}{\frac{K_2}{L}} = \frac{L}{K_1\gamma_1} + \frac{K_1(1-\gamma_1)}{K_1\gamma_1 + L} \leq 1 + 2.$$

Similarly when $\frac{K_2}{L} < \frac{1}{2} \frac{K_1(1-\gamma_1)}{K_1\gamma_1 + L}$, the gap is bounded as

$$G = \frac{\frac{K_2 + K_1(1-\gamma_1)}{K_1\gamma_1 + L}}{\frac{1}{2} \frac{K_1(1-\gamma_1)}{K_1\gamma_1 + L}} = \frac{\frac{K_2}{K_1\gamma_1 + L}}{\frac{1}{2} \frac{K_1(1-\gamma_1)}{K_1\gamma_1 + L}} + 2 \leq 3$$

where the last step considers that $\frac{K_2}{L} < \frac{1}{2} \frac{K_1(1-\gamma_1)}{K_1\gamma_1 + L}$.

This concludes the proof of Theorem 2. \square

APPENDIX C

PROOF OF THEOREM 3

In the considered MISO BC⁶ setting, where d_1, d_2 correspond to the maximum number of cache-aided and cache-less

⁶As mentioned above, this system shares the same fundamental properties with the Interference Channel with cache-aided transmitters and, also, with the wired multi-server setting, thus the following proof is applicable to those settings as well.

users that we can treat, respectively, we have the following trivial bounds, under the assumption of uncoded placement

$$D_L(K_1, \gamma_1, K_2, \gamma_2) \leq l_1 \cdot (K_1\gamma_1 + 1) + l_2 = L + l_1 K_1\gamma_1 \quad (65)$$

where $l_1, l_2 \in \{0, 1, \dots, L\}$ correspond to the number of streams dedicated to each group of users.

In other words, each stream can provide either one DoF to a cache-less user or sum-DoF of $K_1\gamma_1 + 1$ to some cache-aided users. Thus, the minimum transmission time, for a specific set of parameters can be calculated by optimizing variable l_1 as follows

$$T^* \geq \min_{l_1 \in (0, L)} \max \left\{ \frac{K_1(1-\gamma_1)}{l_1(1+K_1\gamma_1)}, \frac{K_2}{L-l_1} \right\} \quad (66)$$

It is obvious that Eq. (66) is minimized when the two quantities are equal, since both are continuous and one is increasing and the other is decreasing. Thus the point that achieves T^* is for $l_1 = \frac{L}{2}$, resulting in the optimal delivery time of

$$T^* = T_{K_1} \frac{\tilde{L}}{L} \quad (67)$$

which is the delivery time in Eq. (7).

APPENDIX D

A NEW CACHE-AIDED MULTI-ANTENNA DELIVERY ALGORITHM

In this section we present a new multi-antenna coded caching algorithm. The presentation is done for the general L -antenna MISO BC channel with K cache-aided users, where each is equipped with a cache of normalized size $\gamma \in (0, 1)$.

The main idea behind the algorithm is to transmit, in each slot, $K\gamma + L$ subfiles, again in an information vector with L messages. We achieve this by creating an L -length vector which is further multiplied by a ZF precoding matrix. The entries of the vector consist of one XOR, comprised of $K\gamma + 1$ subfiles (created exactly as in the algorithm of [1]), and $L - 1$ uncoded subfiles. We continue with the placement and delivery phases.

A. Cache Placement

Initially, each file is subpacketized into $S = \binom{K}{K\gamma}$ subpackets, which are further split into $K\gamma + L$ smaller packets. We will assume that $T_1(K, \gamma) = \frac{K(1-\gamma)}{1+K\gamma}$ is an integer, while extending the scheme to non-integer values requires a slightly increased subpacketization. Users’ caches are filled according to

$$\mathcal{Z}_{k \in [K]} = \{W_\tau^{n, \phi} : \tau \subset [K], |\tau| = K\gamma, k \in \tau, \quad (68) \\ \forall \phi \in [K\gamma + L], \forall n \in [N]\}.$$

The purpose of index ϕ is to guarantee the delivery of “fresh” information, a total of $K\gamma + L$ subfiles for each associated index τ . We will refrain from using this index in the following algorithm, in order to keep the notation clearer, but we will show that data from each subfile is transmitted $K\gamma + L$ times, thus showing that each individual ϕ, τ pair will indeed be transmitted.

B. Delivery Phase

In each delivery slot, as discussed above, we will create a vector of size $L \times 1$, where one of its entries will be a XOR comprized of $K\gamma + 1$ subfiles, while the remaining $L - 1$ entries will be uncoded subfiles. Then, the vector will be multiplied by an $L \times L$ precoder matrix, which is calculated as the normalized inverse of the channel between the L -antenna transmitter and a subset of the $K\gamma + L$ users, namely one of the users of the XOR and the $L - 1$ users that will be the recipients of the uncoded messages. The process is written in the form of a pseudo-code in Alg. 3 and will be further described in the following paragraph. We remind that \mathcal{H}_λ^{-1} denotes the normalized inverse of the channel matrix formed between the L antenna transmitter and the users in set λ , while $\beta_{\tau,k} \subseteq [K] \setminus \tau$ is a set of $L - 1$ elements, which are selected to be the elements following the element $k \in [K] \setminus \tau$.

Algorithm 3: Delivery Phase

```

1 for all  $\chi \subseteq [K], |\chi| = K\gamma + 1$  (pick XOR) do
2   for all  $s \in \chi$  (pick precoded user) do
3     Set:  $\tau = \chi \setminus \{s\}$ 
4     Set:  $\lambda = \{s\} \cup \beta_{\tau,s}$ .
5     Transmit:

```

$$\mathbf{x}_{s,\tau} = \mathcal{H}_\lambda^{-1} \cdot \begin{bmatrix} \bigoplus_{k \in \chi} W_{\chi \setminus \{k\}}^{d_k} \\ W_\tau^{d_{\beta_{\tau,s}(1)}} \\ \vdots \\ W_\tau^{d_{\beta_{\tau,s}(L-1)}} \end{bmatrix} \quad (69)$$

Details of Algorithm 3: The algorithm begins by selecting a subset χ of the users of size $K\gamma + 1$. For these users, the algorithm will form a XOR in the same way as does the algorithm of [1]. Then, the algorithm selects one user, s , from the users of set χ , where this user will be helped by precoding. It is easy to see that the subfile index that this user will receive is $\chi \setminus \{s\} = \tau$.

Further, the remaining $L - 1$ users that are scheduled to receive from the transmitted vector, need to be selected. These users are described by set $\beta_{\tau,s}$, which is calculated by finding the $L - 1$ consecutive elements of set $[K] \setminus \tau$ after element s . For example, if $\chi = \{1, 2, 3\}$, $K = 5$, $L = 2$ and $s = 1$, then $[K] \setminus \tau = \{1, 4, 5\}$ thus, $\beta_{\tau,s} = \{4\}$, as 4 comes right after element $s = 1$. The users of set $\tau \cup \{s\} \cup \beta_{\tau,s}$ are the $L + K\gamma$ users that will receive a subfile in this slot.

For the above selected users, the algorithm creates an $L \times 1$ vector, where one of the elements is a XOR designed for the users in set χ , while the remaining elements correspond to subfiles indexed with τ and intended for the users in set $\beta_{\tau,s}$.

Further, the algorithm forms the precoder matrix \mathcal{H}_λ^{-1} such that it is the normalized inverse of the channel matrix between the L -antenna transmitter and the users in $\lambda = \{s\} \cup \beta_{\tau,s}$.

Finally, the transmitted vector is created by multiplying the precoder matrix with the vector containing the messages.

a) Decoding Process: We begin with the users of set λ i.e., the ‘‘precoding-assisted’’ users. Due to the design of the precoder, we can see that these users will receive either the XORed message (user s) or each of the uncoded messages to the respective user i.e.,

$$y_{k \in \lambda} = \mathbf{h}_k^H \mathbf{x}_{s,\tau} = \begin{cases} \bigoplus_{k \in \chi} W_{\chi \setminus \{k\}}^{d_k}, & k = s \\ W_\tau^{d_k}, & \text{else} \end{cases} \quad (70)$$

where for simplicity we have removed the noise. It is easy to see that users in set $\beta_{\tau,s}$ will be assisted by precoding, thus will only ‘‘see’’ the uncoded subfile that they want. Further, user s will receive XOR X_χ which can proceed to decode using its cached content.

On the other hand, users in set τ will be receiving a linear combination of all L messages, which will proceed to decode using both CSIT knowledge and their cached subfiles. The received message at some user $k \in \tau$ takes the form

$$\begin{aligned} y_{k \in \tau} &= \mathbf{h}_k^H \mathbf{x}_{s,\tau} \\ &= \mathbf{h}_k^H \mathbf{h}_{\lambda \setminus \{s\}}^\perp X_\chi + \mathbf{h}_k^H \sum_{i=1}^{L-1} \mathbf{h}_{\lambda \setminus \beta_{\tau,s}(i)}^\perp W_\tau^{d_{\beta_{\tau,s}(i)}}. \end{aligned} \quad (72)$$

We can see that in Eq. (72), the subfiles that are included in the summation term have all been cached by all receivers of set τ , and as such they can be removed from the equation. What remains is XOR X_χ which, by design, is decodable by all users in τ .

Corollary 5. *In Algorithm 3, each requested subfile $W_\tau^{d_k}, k \in [K]$ is transmitted exactly $K\gamma + L$ times.*

Proof. Since each subfile (W_τ^n) is divided into a total of $K\gamma + L$ smaller subfiles (cf. index ϕ), in this section we aim to show that each of these subfiles will be transmitted exactly once i.e., that each $W_\tau^{d_k}, k \in [K]$ appears in $K\gamma + L$ transmissions.

We split the proof in two steps, where in the first we prove that any requested subfile will be transmitted $K\gamma + 1$ times while combined with other subfiles to form the XOR message, while in the second we prove that any requested subfile is transmitted $L - 1$ times while being a part of the uncoded messages.

First, we can see that each XOR is transmitted a total of $K\gamma + 1$ times, i.e. for each different $s \in \chi$ (Step 2), which implies that requested subfile $W_\tau^{d_k}$ will be transmitted $K\gamma + 1$ times as part of a XOR.

Further, we can see that this subfile can be potentially transmitted, as part of the uncoded elements of the message, when χ is of the form $\chi = \tau \cup \{s\}$, while variable s (Step 2) takes values $s \in [K] \setminus \tau \setminus \{k\}$, thus s can take $K - K\gamma - 1$ different values.

We can discern two cases, namely $K(1 - \gamma) = L$ and $K(1 - \gamma) > L$. In the first case where $K - K\gamma = L$, it is clear that for every $s \in [K] \setminus \tau \setminus \{k\}$ then $\beta_{s,\tau} = [K] \setminus \tau \setminus \{s\}$, thus k will be included in each transmission, which amounts to $L - 1$ different subfiles.

In the second case, where the size of set $[K] \setminus \tau$ is bigger than L , only a subset of the users will be selected every time to form sets $\beta_{s,\tau}, \forall s \in [K] \setminus \tau \setminus \{k\}$. Using $p_k \in \{1, 2, \dots, K - K\gamma\}$, $k \in [K] \setminus \tau$ to denote the position of element k in set $[K] \setminus \tau$, we can see that $k \in \beta_{s,\tau}$ if and only if for some $l \in \{2, \dots, L\}$, the following equality holds

$$p_k = (p_s + l) \pmod{(K - K\gamma) - 1}. \quad (73)$$

The condition of Eq. (73) can be satisfied for exactly $L - 1$ different values of s . Further, for any given s it can only be satisfied by a single l , thus k appears in exactly $L - 1$ sets $\beta_{s,\tau}$, which completes the proof. \square

APPENDIX E

EXTENSION OF THE TWO TYPE CACHE-AIDED SCHEME

In this section we present an extension of Alg. 2 to accommodate any values $L_1, L_2 \in [1, L - 1]$, such that $L_1 + L_2 = L$. The main premise is to increase the per-type subpacketization by some factor $d \in \mathbb{N}$, such that $d \cdot L_1 \in \mathbb{N}, d \cdot L_2 \in \mathbb{N}$. Hence the total subpacketization becomes

$$S = d^2(L_1 + K_1\gamma_1) \binom{K_1}{K_1\gamma_1} (L_2 + K_2\gamma_2) \binom{K_2}{K_2\gamma_2}. \quad (74)$$

The new scheme works by repeating d^2 times Alg. 2, with the difference that some transmissions will allocate $\lceil L_1 \rceil$ streams to users of set \mathcal{K}_1 and at the same time will allocate $\lceil L_2 \rceil$ streams to set \mathcal{K}_2 and in some transmissions will allocate $\lceil L_1 \rceil$ streams to users of set \mathcal{K}_1 and at the same time will allocate $\lceil L_2 \rceil$ streams to set \mathcal{K}_2 .

This way, it allows the average allocation of L_1 and L_2 streams to each user type, which leads to the DoF $D_L(K_1, \gamma_1, K_2, \gamma_2) = L + K_1\gamma_1 + K_2\gamma_2$.

REFERENCES

- [1] M. A. Maddah-Ali and U. Niesen, "Fundamental limits of caching," *IEEE Transactions on Information Theory*, vol. 60, pp. 2856–2867, May 2014.
- [2] E. Lampsiris and P. Elia, "Full coded caching gains for cache-less users," in *IEEE Information Theory Workshop (ITW)*, pp. 1–5, Nov 2018.
- [3] S. P. Shariatpanahi, S. A. Motahari, and B. H. Khalaj, "Multi-server coded caching," *IEEE Transactions on Information Theory*, vol. 62, pp. 7253–7271, Dec 2016.
- [4] N. Naderializadeh, M. A. Maddah-Ali, and A. S. Avestimehr, "Fundamental limits of cache-aided interference management," *IEEE Transactions on Information Theory*, vol. 63, pp. 3092–3107, May 2017.
- [5] E. Lampsiris and P. Elia, "Adding transmitters dramatically boosts coded-caching gains for finite file sizes," *IEEE Journal on Selected Areas in Communications (JSAC)*, vol. 36, pp. 1176–1188, June 2018.
- [6] E. Lampsiris and P. Elia, "Achieving full multiplexing and unbounded caching gains with bounded feedback resources," in *IEEE International Symposium on Information Theory (ISIT)*, pp. 1440–1444, June 2018.
- [7] E. Lampsiris and P. Elia, "Bridging two extremes: Multi-antenna coded caching with reduced subpacketization and CSIT," in *IEEE 20th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, pp. 1–5, July 2019.
- [8] S. P. Shariatpanahi, G. Caire, and B. Hossein Khalaj, "Physical-layer schemes for wireless coded caching," *IEEE Transactions on Information Theory*, vol. 65, pp. 2792–2807, May 2019.
- [9] A. Tölli, S. P. Shariatpanahi, J. Kaleva, and B. Khalaj, "Multi-antenna interference management for coded caching," *arXiv preprint arXiv:1711.03364*, 2017.
- [10] E. Lampsiris, P. Elia, and G. Caire, "Bridging the gap between multiplexing and diversity in finite SNR multiple antenna coded caching," *To appear in Asilomar Conference on Signals, Systems, and Computers*, 2019.
- [11] M. Bayat, R. K. Mungara, and G. Caire, "Achieving spatial scalability for coded caching over wireless networks," *arXiv preprint arXiv:1803.05702*, 2018.
- [12] K. Ngo, S. Yang, and M. Kobayashi, "Scalable content delivery with coded caching in multi-antenna fading channels," *IEEE Transactions on Wireless Communications*, vol. 17, pp. 548–562, Jan 2018.
- [13] J. Zhang and P. Elia, "Fundamental limits of cache-aided wireless BC: Interplay of coded-caching and CSIT feedback," *IEEE Transactions on Information Theory*, vol. 63, pp. 3142–3160, May 2017.
- [14] X. Yi and G. Caire, "Topological coded caching," in *2016 IEEE International Symposium on Information Theory (ISIT)*, pp. 2039–2043, July 2016.
- [15] A. Sengupta, R. Tandon, and O. Simeone, "Cache aided wireless networks: Tradeoffs between storage and latency," in *2016 Annual Conference on Information Science and Systems (CISS)*, pp. 320–325, March 2016.
- [16] Y. Cao, M. Tao, F. Xu, and K. Liu, "Fundamental storage-latency trade-off in cache-aided MIMO interference networks," *IEEE Transactions on Wireless Communications*, vol. 16, pp. 5061–5076, Aug 2017.
- [17] J. S. P. Roig, D. Gündüz, and F. Tosato, "Interference networks with caches at both ends," in *2017 IEEE International Conference on Communications (ICC)*, pp. 1–6, May 2017.
- [18] J. Zhang, F. Engelmann, and P. Elia, "Coded caching for reducing CSIT-feedback in wireless communications," in *2015 53rd Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pp. 1099–1105, Sep. 2015.
- [19] J. Zhang and P. Elia, "Wireless coded caching: A topological perspective," in *IEEE International Symposium on Information Theory (ISIT)*, pp. 401–405, June 2017.
- [20] L. Tang and A. Ramamoorthy, "Coded caching for networks with the resolvability property," in *2016 IEEE International Symposium on Information Theory (ISIT)*, pp. 420–424, July 2016.
- [21] E. Lampsiris, J. Zhang, and P. Elia, "Cache-aided cooperation with no CSIT," in *IEEE International Symposium on Information Theory (ISIT)*, pp. 2960–2964, June 2017.
- [22] A. Sengupta, R. Tandon, and T. C. Clanc, "Layered caching for heterogeneous storage," in *2016 50th Asilomar Conference on Signals, Systems and Computers*, pp. 719–723, Nov 2016.
- [23] M. Mohammadi Amiri, Q. Yang, and D. Gündüz, "Decentralized caching and coded delivery with distinct cache capacities," *IEEE Transactions on Communications*, vol. 65, pp. 4657–4669, Nov 2017.
- [24] D. Cao, D. Zhang, P. Chen, N. Liu, W. Kang, and D. Gündüz, "Coded caching with heterogeneous cache sizes and link qualities: The two-user case," *arXiv preprint arXiv:1802.02706*, 2018.
- [25] A. M. Ibrahim, A. A. Zewail, and A. Yener, "Centralized coded caching with heterogeneous cache sizes," in *2017 IEEE Wireless Communications and Networking Conference (WCNC)*, pp. 1–6, March 2017.
- [26] A. M. Ibrahim, A. A. Zewail, and A. Yener, "Device-to-device coded caching with heterogeneous cache sizes," in *2018 IEEE International Conference on Communications (ICC)*, pp. 1–6, IEEE, 2018.
- [27] A. M. Daniel and W. Yu, "Optimization of heterogeneous coded caching," *arXiv preprint arXiv:1708.04322*, 2017.
- [28] B. Asadi, L. Ong, and S. J. Johnson, "Centralized caching with unequal cache sizes," in *IEEE Inf. Theory Workshop (ITW)*, Nov 2018.
- [29] A. M. Ibrahim, A. A. Zewail, and A. Yener, "Coded caching for heterogeneous systems: An optimization perspective," *IEEE Transactions on Communications*, vol. 67, pp. 5321–5335, Aug 2019.
- [30] A. M. Ibrahim, A. A. Zewail, and A. Yener, "Benefits of coded placement for networks with heterogeneous cache sizes," in *2018 52nd Asilomar Conference on Signals, Systems, and Computers*, pp. 1604–1608, Oct 2018.
- [31] K. Wan, D. Tuninetti, and P. Piantanida, "On the optimality of uncoded cache placement," in *IEEE Information Theory Workshop (ITW)*, pp. 161–165, Sep. 2016.
- [32] Q. Yu, M. A. Maddah-Ali, and A. S. Avestimehr, "Characterizing the rate-memory tradeoff in cache networks within a factor of 2," *IEEE Transactions on Information Theory*, vol. 65, pp. 647–663, Jan 2019.
- [33] G. Paschos, E. Bastug, I. Land, G. Caire, and M. Debbah, "Wireless caching: technical misconceptions and business barriers," *IEEE Communications Magazine*, vol. 54, pp. 16–22, August 2016.

- [34] G. Aagnarsson and R. Greenlaw, *Graph theory: Modeling, applications, and algorithms*. Prentice-Hall, Inc., 2007.
- [35] K. Fukuda and T. Matsui, "Finding all the perfect matchings in bipartite graphs," *Applied Mathematics Letters*, vol. 7, no. 1, pp. 15 – 18, 1994.
- [36] M. Li, L. Ong, and S. J. Johnson, "Improved bounds for multi-sender index coding," in *IEEE International Symposium on Information Theory (ISIT)*, pp. 3060–3064, June 2017.
- [37] E. Parrinello, A. Ünsal, and P. Elia, "Coded caching with shared caches: Fundamental limits with uncoded prefetching," *arXiv preprint arXiv:1809.09422*, 2018.