Low-Complexity MISO Cache-Aided Communication via Meta-User Scheduling

Soheil Mohajer, *Member, IEEE*, and Itsik Bergel, *Senior Member, IEEE*

***Abstract*—We present a novel low-complexity scheme for cache-aided communication, where a multi-antenna base station serves multiple single-antenna mobile users. The scheme is based on dividing the users into *meta-users*, where all users in the same meta-user store the same content during the placement phase. The inter meta-user interference is mitigated by using the cache as well as zero forcing, while the interference between users of the same meta-user is mitigated by zero forcing. Compared to the current state of the art, this scheme is feasible for a wider range of parameters. Moreover, while still achieving the optimal number of degrees of freedom (DoF), the proposed scheme imposes the same or less complexity compared to all the known schemes for each set of parameters. Consequently, the proposed scheme enables practical implementation of cache-aided communication for a large number of users.**

***Index Terms*—cache-aided communication, MISO, subpacketization level, complexity**

I. Introduction

The performance of wireless communication has been improved significantly by development of new technologies such as MIMO systems, mmWave, etc. However, considering the overwhelming growth in demand and the shift of the data usage from voice to content delivery, we need to fully exploit all the existing resources to be able to keep up with the demand. In particular, caching techniques allow to benefit from the off-peak hours of networks and the time variant nature of the traffic. In a nut shell, cache aided communication allows us to shift a part of the traffic from the peak hours to lower traffic time of the network.

The gain of traditional caching is limited to the fraction of the database stored at each individual user, which is typically negligible. In contrast, the so called *coded caching* or *cache-aided communication* introduced in [1] allow for two separate gains: (1) a *local gain* due to the fraction of the desired data of each user which is cached by that user and (2) a *global gain* due to the interference cancellation and broadcasting opportunity provided by caching the desired data of one user by other users. While the local gain is still small, the global gain scales with the aggregate size of the cache distributed among all the users in the network.

The scheme consists of a placement phase and a delivery phase [1]. During the placement phase, prior to knowing the users' requests, we can pre-fetch and store at each users' memory some packets from the files in the database. Once the requests are revealed, the server generates a set of coded messages and transmits them to all the receivers during the delivery phase. All users should be able to decode their desired file from the received signal and their cache content.

The key feature of coded caching is the feasibility of *multicasting packets*, which are combination of segments of multiple files. Such packets are carefully designed at the transmitter so that each intended receiver has all the interfering parts in its cache, and is able to extract its own desired information from the combination. This leads to a an achievable degrees of freedom (DoF), proportional to the number of copies of each piece of data cached in the system.

In a multiple-input single-output (MISO) network where the transmitter is equipped with L transmit antennas, L users can be simultaneously served, and thus L degrees of freedom (DoFs) can be achieved. Shariatpanahi *et al.* showed that coded caching can achieve $L+M$ DoFs in a broadcast system with L transmit antennas at the server and an aggregate cache size that distributedly stores M copies of the database across the users [2].

However, the scheme of Shariatpanahiet al. (which we refer to as the SCK scheme) suffers from a very high implementation complexity. This is mostly due to its huge subpacketization level. More precisely, the SCK scheme requires dividing each file into $\binom{U}{M}\binom{U-M-1}{L-1} = O\left(U^{M+L-1}\right)$ file segments, where U is the number of users in the network [2]. Due to its fast growing subpacketization level, the SCK scheme is only feasible for networks with very small number of users or applications with very large files (which can be divided to sufficiently large number of segments).

The problem of subpacketization in cache-aided communication was widely studied for the single antenna setting. In particular, the problem was formulated as an optimization problem in [3], [4]. Moreover, some combinatorial solutions were proposed for specific range of parameters [5], [6].

However, in the MISO setting, the exponent of subpacketization order not only increases by the cache size, but also by the number of antennas. In a seminal work [7], Lampiris and Elia proposed the a placement and delivery scheme based on grouping and cache replication ideas. Their scheme (termed herein the LE scheme) treats the network as if there are only U/L *effective* users, and hence requires much lower subpacketization level. The LE scheme can achieve the same $M+L$ DoFs as the SCK scheme, but, only if the aggregate number of copies of the database is divisible by the number of antennas, i.e., $L|M$. This is a significant drawback, as in practice, M is typically small.

S. Mohajer is with the Department of Electrical and Computer Engineering, University of Minnesota, Twin Cities, MN 55455, USA, (email: soheil@umn.edu). I. Bergel is with the Faculty of Engineering, at Bar-Ilan University, Ramat Gan 52900, Israel (email: itsik.bergel@biu.ac.il).

The work of S. Mohajer is supported by the National Science Foundation under grant CCF-1749981.

ISBN 978-3-8007-5686-5

An alternative approach based on scheduling is introduced in [8]–[10], that also achieves DoF of $M+L$. In contrast to the SCK scheme, where $\binom{M+L}{M+1}$ groups of size $M+1$ are transmitted in each time block, the scheduling scheme targets only $g=\frac{M+L}{M+1}$ groups to be served. This scheme is simpler if g is integer [8] but was also generalized to non-integer g [9]. Thus, the scheduling scheme requires a subpacketization level of only $\binom{U}{M}=O\left(U^M\right)$. However, the subpacketization level of the scheduling scheme is higher than that of the LE scheme when M is large and $L|M$.

In this paper, we present a novel scheme that combines the benefits of the LE scheme and the scheduling scheme. Similar to the LE scheme, the proposed scheme groups the users into meta-users. While the size of meta-user size in the LE scheme is limited to L, our scheme is flexible and allows for various sizes of meta-users. Furthermore, our scheme is applicable for any (integer) value of M and L, while still achieving $M+L$ DoFs. The scheme requires a subpacketization level lower than or equal to that of the LE scheme and the scheduling scheme for any parameter values.

The rest of this paper is organized as follows: the system model is introduced in Section II, while Section III gives the relevant details on the LE and scheduling schemes. Our proposed scheme is presented in Section IV, followed by an analysis of the subpacketization level in Section V.

Notation. Throughout this paper, $[a:b]$ denotes the set $\{a,a+1,\ldots,b\}$, and $[b]=\{1,2,\ldots,b\}$, for $a,b\in\mathbb{Z}$. For two integers n and k we have $\binom{n}{k}=\frac{n!}{k!(n-k)!}$. We use bold letters (e.g. $\mathbf{h}$) to denote vectors. The average is denoted by $\mathbb{E}[\cdot]$, and all the logarithms are in base 2.

II. System Model

We consider a network with U users each equipped with one receive antenna, and a single base-station (BS) which is equipped with L transmit antennas. We focus on a wide-band communication scenario, in which the bandwidth, B, is divided into K frequency bins (e.g., OFDM), and each bin $k\in[K]$ carries one modulated symbol at a time, without inter-symbol interference. The received sample after matched filtering for the kth frequency bin at user u is given by

$$y_{k,u}(t)=\mathbf{h}_{k,u}\mathbf{x}_k(t)+z_{k,u}(t), \tag{1}$$

where $\mathbf{x}_k(t)\in\mathbb{C}^{L\times 1}$ is the transmit vector at time block t over the frequency bin k, $z_{k,u}\sim\mathbb{CN}\left(0,B/K\right)$ is the additive white Gaussian noise sample at user u in the frequency bin k, and $\mathbf{h}_{k,u}\in\mathbb{C}^{1\times L}$ is the channel vector from the BS to user u in the frequency bin k. We assume the BS has a total power constraint of P, and the perfect channel state information (CSI) is available at the BS. Moreover, we consider a *homogeneous* system, in which each user experiences a fading channel, where the channel statistics of all users are identical. More precisely, we assume $\mathbf{h}_{k,u}\sim\mathbb{CN}(0,1)$.

The BS has a dictionary of N files, namely $\{W_1,W_2,\ldots,W_N\}$, each of size F bits. Each user is interested in one of the files, chosen uniformly at random. In a cache-aided communication system, each user $u\in[U]$ is equipped with a memory Z_u that can store up to MNF/U bits. Hence, a total of M copies of the entire dictionary of the files can be distributedly stored across the users.

Cache *placement*, the process of filling the storage of the users with partial information from the dictionary, takes place before the users' demands are revealed. Later, each user u requests one file from the dictionary, namely W_{d_u}, and the BS starts serving the users during the *delivery phase*. At the end of the delivery phase, each user u should be able to decode W_{d_u} from Z_u and the signal received from the BS.

Throughout this paper, for a group of users $\mathcal{A}\subseteq[U]$ with $|\mathcal{A}|\leq L-1$, we use $\mathbf{h}_k^{\perp}[\mathcal{A}]\in\mathbb{C}^{L\times 1}$ to denote a unit-length beamforming vector which is orthogonal to the channel vectors of all users in $\mathcal{A}$ in frequency bin k. That is,

$$\begin{aligned}\mathbf{h}_{k,u}\mathbf{h}_k^{\perp}[\mathcal{A}]&=0, \qquad \forall u\in\mathcal{A},\\ \|\mathbf{h}_k^{\perp}[\mathcal{A}]\|&=1.\end{aligned} \tag{2}$$

Note that such a vector is unique when $|\mathcal{A}|=L-1$ and the channel to the users in $\mathcal{A}$ are linearly independent. Otherwise, $\mathbf{h}_k^{\perp}[\mathcal{A}]$ refers to any vector satisfying (2).

III. The Existing Schemes

In this section we review the Lampiris and Elia scheme (referred to as the LE Scheme) [7], which is the state-of-the-art for low-complexity transmission over cache-aided MISO communication networks. We also review the novel scheduling approach that we recently introduced [8]–[10] (referred to as the scheduling scheme), which allows complexity reduction by scheduling transmissions for groups of user.

A. The LE Scheme

The key idea in this scheme is grouping users into a number of *meta-users*. To this end, we first split U users into $\mathbb{U}=U/L$ *meta-users*, each including L actual users. We denote the set of meta-users by $[\mathbb{U}]=\{V_1,V_2,\ldots,V_{\mathbb{U}}\}$. Without loss of generality, we may assume meta-user V_j consists of the jth group of users, that is,

$$V_j=\{u:\lceil u/L\rceil=j\}=\{(j-1)L+1,(j-1)L+2,\ldots,jL\}.$$

We also define the normalized cache parameter $\mathbb{M}=M/L$. Then each file W_n is split into $\binom{\mathbb{U}}{\mathbb{M}}$ segments, each indexed by a set $\mathbb{S}$ of size $|\mathbb{S}|=\mathbb{M}$ from the set $\{1,2,\ldots,\mathbb{U}\}$, that is,

$$W_n=\{W_n^{\mathbb{S}}:\mathbb{S}\subseteq[\mathbb{U}],|\mathbb{S}|=\mathbb{M}\},\qquad n\in[N].$$

Then, the cache content of user u will be

$$Z_u=\bigcup_{n\in[N]}\{W_n^{\mathbb{S}}:\lceil u/L\rceil\in\mathbb{S}\}. \tag{3}$$

It is clear from (3), we have $Z_u=Z_v$ for every pair of users u and v which belong to the same meta-user, or equivalently, $\lceil u/L\rceil=\lceil v/L\rceil$. Moreover, it is easy to verify that

$$|Z_u|=N\binom{\mathbb{U}-1}{\mathbb{M}-1}\frac{F}{\binom{\mathbb{U}}{\mathbb{M}}}=\frac{N\mathbb{M}}{\mathbb{U}}F=\frac{NM}{U}F.$$

The time blocks of the delivery phase are indexed by sets $\mathcal{Q}\subseteq[\mathbb{U}]$ with $|\mathcal{Q}|=\mathbb{M}+1$. Every user u with $\lceil u/L\rceil\in\mathcal{Q}$

ISBN 978-3-8007-5686-5

will be served during the time block $\mathcal{Q}$. Therefore, a total of $L|\mathcal{Q}| = L(\mathbb{M}+1) = M+L$ users will be served in each time block.

Consider a (actual) user v and a time block $\mathcal{Q}$ with $v \in V \in \mathcal{Q}$ (i.e., the meta-user V that includes v is served in time block $\mathcal{Q}$), who has requested file W_{d_v}. During the time block $\mathcal{Q}$ we serve this user by a segment of its requested file, namely, $W_{d_v}^{\mathcal{Q}\setminus\{V\}}$. This file segment will be encoded to $\mathbf{w}_{d_v}^{\mathcal{Q}\setminus\{V\}}$ of length $K\tau^{\mathsf{LE}}$. Such a codeword will be further divided into K chunks $\mathbf{w}_{d_v,k}^{\mathcal{Q}\setminus\{V\}}$ each of length τ^{LE}, for $k \in [K]$. The transmit signal in time block $\mathcal{Q}$ in frequency bin $k \in [K]$ will be then determined by

$$\mathbf{X}_k[\mathcal{Q}] = p \sum_{V\in\mathcal{Q}} \sum_{v\in V} \mathbf{h}_k^{\perp}[V \setminus \{v\}]\, \mathbf{w}_{d_v,k}^{\mathcal{Q}\setminus\{V\}}.$$

where $p = \sqrt{\frac{P}{K(M+L)}}$ is the power allocated for each message in the frequency bin k.

The received signal at an active user u in a meta-group $V_\circ \in \mathcal{Q}$ will be

$$\begin{aligned}\mathbf{y}_{u,k}[\mathcal{Q}] &= \mathbf{h}_{u,k}\mathbf{X}_k[\mathcal{Q}] + \mathbf{z}_{u,k}[\mathcal{Q}] \\ &= p \sum_{V\in\mathcal{Q}} \sum_{v\in V} \mathbf{h}_{u,k}\mathbf{h}_k^{\perp}[V \setminus \{v\}]\, \mathbf{w}_{d_v,k}^{\mathcal{Q}\setminus\{V\}} + \mathbf{z}_{u,k}[\mathcal{Q}].\end{aligned} \tag{4}$$

Consider a meta-user $V \neq V_\circ$. Recall that since $u \notin V$, then we have $\lceil u/L \rceil = V_\circ \in \mathcal{Q} \setminus \{V\}$, and hence, the cache placement strategy in (3) implies that all the file segments of form $W_n^{\mathcal{Q}\setminus\{V\}}$ are cached at user u. Hence, the user can compute the associated codeword chuck $\mathbf{w}_{n,k}^{\mathcal{Q}\setminus\{V\}}$ and subtract the corresponding term from $\mathbf{y}_{u,k}[\mathcal{Q}]$. This yields to

$$\begin{aligned}&\mathbf{y}_{u,k}[\mathcal{Q}] - p \sum_{\substack{V\in\mathcal{Q}\\ V\neq V_\circ}} \sum_{v\in V} \mathbf{h}_{u,k}\mathbf{h}_k^{\perp}[V \setminus \{v\}]\, \mathbf{w}_{d_v,k}^{\mathcal{Q}\setminus\{V\}} \\ &\quad= p \sum_{v\in V_\circ} \mathbf{h}_{u,k}\mathbf{h}_k^{\perp}[V_\circ \setminus \{v\}]\, \mathbf{w}_{d_v,k}^{\mathcal{Q}\setminus\{V_\circ\}} + \mathbf{z}_{u,k}[\mathcal{Q}] \\ &\quad\overset{(a)}{=} p\mathbf{h}_{u,k}\mathbf{h}_k^{\perp}[V_\circ \setminus \{u\}]\, \mathbf{w}_{d_u,k}^{\mathcal{Q}\setminus\{V_\circ\}} + \mathbf{z}_{u,k}[\mathcal{Q}],\end{aligned}$$

where (a) holds since for every $v \neq u$ we have $u \in V_\circ \setminus \{v\}$, and hence, (2) implies that $\mathbf{h}_{u,k}\mathbf{h}_k^{\perp}[V_\circ \setminus \{v\}] = 0$. Therefore, upon receiving $\{\mathbf{y}_{u,k}[\mathcal{Q}]\}_{k=1}^K$ and cancelling the interference, user u can decode file segment $W_{d_u}^{\mathcal{Q}\setminus\{V_\circ\}}$. Collecting all such segments for all communication blocks $\mathcal{Q}$ (with $\mathcal{Q} \subseteq [\mathbb{U}]$ and $|\mathcal{Q}| = \mathbb{M}+1$) and those caches in Z_u, user u can completely retrieve its desired file W_{d_u}.

The main advantage of this scheme is its subpacketization level $\binom{\mathbb{U}}{\mathbb{M}} = \binom{U/L}{M/L}$, which significantly smaller than that of the SCK scheme. Moreover, the number of time blocks required to serve all users is only $\binom{\mathbb{U}}{\mathbb{M}+1} = \binom{U/L}{M/L+1}$. It is worth noting the length of communication blocks in the LE and the SCK schemes are different. However, for the feasibility of this scheme, it is required that $\mathbb{U} = U/L$ and $\mathbb{M} = M/L$ be integer numbers. While the first conditions can be fulfilled by a paying small penalty (by appending less than L dummy users so that the total number of users becomes an integer multiple of L), the second requirement is stringent, and turns out to be the major shortcoming of the LE scheme.

B. The Scheduling Scheme

The placement strategy for this case is similar to that of [1] and [2]. In particular, we first split each file W_n into $\binom{U}{M}$ segments, and label them as W_n^S where S is any subset of $[U]$ of size M. Then, the cache content of user is given by

$$Z_u = \bigcup_{n\in[N]} \{W_n^S : u \in S\}. \tag{5}$$

The delivery phase is divided into T time blocks. In each time block we can serve *up to* $M+L$ users. Serving users in each time block is performed by serving *groups* of size $M+1$. Therefore, the number of groups to be served in each time block is given by $g \triangleq \frac{M+L}{M+1}$. In the this paper we only consider scenarios with integer values of g. Scheduling for non-integer values of g was presented in [9], and its implementation of the proposed scheme for the case of non-integer g is presented in [10].

Consider all subsets of $[U]$ of size $M+1$, that is, $\mathcal{G} \triangleq \{\mathcal{B} \subseteq [U] : |\mathcal{B}| = M+1\}$. In each time block m, we can serve g of such subsets of users (referred to as *group*). The properties of groups that can be served together are summarized in the following definition.

Definition 1 (Valid Scheduling)**.** *For integer value of g, a* scheduling *is an array $\mathcal{T}$ of length $T^\star$, given by*

$$\mathcal{T} = (\mathcal{T}[1], \mathcal{T}[2], \ldots, \mathcal{T}[T^\star]),$$

where each element $\mathcal{T}[m]$ is a collection of at most g groups from $\mathcal{G}$, i.e., $\mathcal{T}[m] \subset \mathcal{G}$ with $|\mathcal{T}[m]| \leq g$. Such a scheduling is called valid if and only if

(i) All groups are covered by $\mathcal{T}$, i.e., $\bigcup_{m=1}^{T^\star} \mathcal{T}[m] = \mathcal{G}$.
(ii) The set of groups associated to each time block are disjoint, i.e., $\mathcal{B} \cap \mathcal{B}' = \varnothing$, for every m and every $\mathcal{B}, \mathcal{B}' \in \mathcal{T}[m]$.

For a time block $m \in \{1, 2, \ldots, T^\star\}$, the scheduling $\mathcal{T}[m]$ determines the set of groups to be served in time block m. The length of the scheduling, $T^\star$, determines the duration of the delivery phase. The existence of a good scheduling, that guarantees the optimal DoF for large enough number of users, is proved in [10].

The set of active users in time block m is given by

$$\mathcal{U}[m] = \bigcup\nolimits_{\mathcal{B}\in\mathcal{T}[m]} \mathcal{B}. \tag{6}$$

For every user $v \in \mathcal{B} \in \mathcal{T}[m]$, the file segment $W_{d_v}^{\mathcal{B}\setminus\{v\}}$ is sent in time block m. To this end, we first modulate this file segment to a codeword $\mathbf{w}_{d_v}^{\mathcal{B}\setminus\{v\}}$ of length $\tau^\star K$. We further split the codeword into K chunks, namely $\mathbf{w}_{d_v,k}^{\mathcal{B}\setminus\{v\}}$ for $k \in [K]$, each of length $\tau^\star$, and send each chunk in one frequency bin. The transmit signal for time block m in frequency bin k will be formed by

$$\mathbf{X}_k[m] = p_m \sum_{\mathcal{B}\in\mathcal{T}[m]} \sum_{v\in\mathcal{B}} \mathbf{h}_k^{\perp}[\mathcal{U}[m] \setminus \mathcal{B}]\, \mathbf{w}_{d_v,k}^{\mathcal{B}\setminus\{v\}}, \tag{7}$$

where $p_m = \sqrt{\frac{P}{K\cdot|\mathcal{U}[m]|}} = \sqrt{\frac{P}{K(M+1)\cdot|\mathcal{T}[m]|}}$ is the power allocated to each message in time block m, $\mathbf{X}_k[m] \in \mathbb{C}^{L\times\tau^\star}$

ISBN 978-3-8007-5686-5

and its (ℓ, t) entry will be sent at time block t of time block m over the ℓth transmit antenna.

The received vector at user u in group $\mathcal{B}_\circ \in \mathcal{T}[m]$ in frequency bin k is given by

$$\begin{aligned}\mathbf{y}_{u,k}[m] &= \mathbf{h}_{u,k}\mathbf{X}_k[m] + \mathbf{z}_{u,k}[m] \\ &= p_m \sum_{v\in\mathcal{B}_\circ} \mathbf{h}_{u,k}\mathbf{h}_k^{\perp}[\mathcal{U}[m]\setminus\mathcal{B}_\circ]\,\mathbf{w}_{d_v,k}^{\mathcal{B}_\circ\setminus\{v\}} + \mathbf{z}_{u,k}[m],\end{aligned} \tag{8}$$

where the second equality holds since $\mathbf{h}_{u,k}$ is orthogonal to $\mathbf{h}_k^{\perp}[\mathcal{U}[m]\setminus\mathcal{B}]$ for every $\mathcal{B}\neq\mathcal{B}_\circ$. Note that

$$|\mathcal{U}[m]\setminus\mathcal{B}| = |\mathcal{U}[m]| - |\mathcal{B}| \le g(M+1)-(M+1) = L-1,$$

and hence the beamforming vector $\mathbf{h}_k^{\perp}[\mathcal{U}[m]\setminus\mathcal{B}]$ is well-defined. Moreover, since $u\in\mathcal{B}_\circ$, all file segments $W_n^{\mathcal{B}_\circ\setminus\{v\}}$ (with $v\neq u$) are stored in the cache of user u, and hence the corresponding interference in (8) can be subtracted from the received signal. Therefore, user u obtains $\mathbf{h}_{u,k}\mathbf{h}_k^{\perp}[\mathcal{U}[m]\setminus\mathcal{B}_\circ]\,\mathbf{w}_{d_u,k}^{\mathcal{B}_\circ\setminus\{u\}} + \mathbf{z}_{u,k}[m]$ for all $k\in[K]$, from which $W_{d_u}^{\mathcal{B}_\circ\setminus\{u\}}$ can be decoded. Lastly, user u can retrieve its desired file W_{d_u} by collecting all its cached and decoded segments.

The scheduling scheme imposes a subpacketization level of $\binom{U}{M}$. Even though the subpacketization is substantially reduced compared to the SCK scheme, it is still significantly higher than that of the LE scheme for systems with a large number of users, U.

IV. The Proposed Scheme

Both the LE scheme and the scheduling scheme can significantly reduce the subpacketization level, while achieving the best known DoF. The limitation of the LE scheme is that it only works if M is divisible by L. This is a significant drawback, as it limits applicability of the LE scheme to specific memory sizes. Furthermore, M (the number of copies of the database distributively cached across the users) is typically a small number, while with the popularity of massive MIMO, we witness an increasing trend in L (the number of antennas at the transmitter).

Here, we propose a new scheme, that generalizes the LE scheme, by allowing more versatile choice of the meta-user size. A key parameter in our proposed scheme is the size of meta-user, which is an integer κ that divides both M and L. Thus, $1\le\kappa\le\gcd(M,L)$, where $\gcd(\cdot,\cdot)$ indicated the greatest common divisor. Our scheme is a generalization of the LE scheme, where it allows for any meta-user size rather than $\kappa = L$. On the other hand, it generalizes the scheduling scheme by scheduling meta-users (instead of actual users).

Consider a set of system parameters (U, M, L) with some meta-user size κ that divides both L and M. We also assume that κ divides U.[1] Similar to the LE scheme, we split the users into $\mathbb{U} = U/\kappa$ meta-users, each including κ users. Let $[\mathbb{U}] = \{V_1,\dots,V_{\mathbb{U}}\}$ denote the set of meta-users, and without loss of generality assume

$$V_j = \{u : \lceil u/\kappa\rceil = j\} = \{(j-1)\kappa+1,\dots,j\kappa\}.$$

[1]Otherwise, we append at most $\kappa-1$ dummy users to the system so that the total number of users become divisible by κ.

Let $\mathbb{M} = M/\kappa$. Each file will be then partitioned into $\binom{\mathbb{U}}{\mathbb{M}}$ segments, and its segments are labeled by subsets of $[\mathbb{U}]$ of size $\mathbb{M}$:

$$W_n = \{W_n^{\mathbb{S}} : \mathbb{S}\subseteq[\mathbb{U}], |\mathbb{S}| = \mathbb{M}\}, \qquad n\in[N].$$

Then, the cache content of user u will be

$$Z_u = \bigcup\nolimits_{n\in[N]} \{W_n^{\mathbb{S}} : \lceil u/\kappa\rceil\in\mathbb{S}\}. \tag{9}$$

It is easy to verify that $|Z_u| = \frac{NM}{U}F$ satisfies the cache size constraint.

After the placement phase is complete, each user u reveals its request, which is a file with index $d_u\in[N]$. We aim to serve $M+L$ users in each time block. These users will be the members of $\mathbb{M}+\mathbb{L}$ meta-users, where $\mathbb{L} = L/\kappa$. We further split these meta-users into groups, where each group includes $\mathbb{M}+1$ meta-users.

Let us define $g = \frac{\mathbb{M}+\mathbb{L}}{\mathbb{M}+1} = \frac{M+L}{M+\kappa}$. In each time block, we serve g groups of meta-users. As mentioned earlier, we only consider the case of integer g in this paper, we refer to [9] for the presentation of the scheme and its implementation for non-integer values of g.

Let $\mathcal{T}$ be a valid scheduling over $[\mathbb{U}]$, as defined in Definition 1. The set of meta-users to be served in time block m is determined by

$$\mathcal{X}[m] = \bigcup\nolimits_{\mathcal{B}\in\mathcal{T}[m]} \mathcal{B},$$

and hence, the set of users to be served in this time block can be found from

$$\mathcal{U}[m] = \bigcup\nolimits_{\mathcal{B}\in\mathcal{T}[m]} \mathcal{U}_B = \bigcup_{\mathcal{B}\in\mathcal{T}[m]} \bigcup\nolimits_{V\in\mathcal{B}} V, \tag{10}$$

where $\mathcal{U}_\mathcal{B} = \bigcup_{V\in\mathcal{B}} V$ is the set of users covered by the group of meta-users $\mathcal{B}$.

Consider a time block $m\in[T^\star]$ and the set of active users in (10). In this time block a user $v\in\mathcal{U}[m]$ with $v\in V\in\mathcal{B}$ for some $\mathcal{B}\in\mathcal{T}[m]$ will be served by the file segment $W_{d_u}^{\mathcal{B}\setminus\{V\}}$. To this end, we generate a transmit signal

$$\mathbf{X}_k[m] = p_m \sum_{\mathcal{B}\in\mathcal{T}[m]} \sum_{V\in\mathcal{B}} \sum_{v\in V} \mathbf{h}_k^{\perp}[\mathcal{N}(\mathcal{U}[m],\mathcal{B},V,v)]\,\mathbf{w}_{d_v,k}^{\mathcal{B}\setminus\{V\}} \tag{11}$$

where

$$\mathcal{N}(\mathcal{U}[m],\mathcal{B},V,v) := (\mathcal{U}[m]\setminus\mathcal{U}_\mathcal{B})\cup(V\setminus\{v\}). \tag{12}$$

In other words, a codeword chunk $\mathbf{w}_{d_v,k}^{\mathcal{B}\setminus\{V\}}$ in (11) will be sent orthogonal to the channel of all users in $\mathcal{N}(\mathcal{U}[m],\mathcal{B},V,v)$, i.e., $\mathbf{w}_{d_v,k}^{\mathcal{B}\setminus\{V\}}$ will be zero-forced at every (active) user that belongs to a *different group of meta-users* as well as those users who belong the *same user* (except v).

Note that

$$\begin{aligned}|\mathcal{N}(\mathcal{U}[m],\mathcal{B},V,v)| &\le \kappa|\mathcal{X}[m]\setminus\mathcal{B}| + |V| - 1 \\ &\le \kappa(\mathbb{L}-1)+\kappa-1 = L-1.\end{aligned}$$

This implies each file segment need to be zero-forced in at most $L-1$ users, which is feasible using L transmit antennas.

The received signal at user $u \in V_\circ \in \mathcal{B}_\circ \in \mathcal{T}[m]$ in frequency bin k can be written as

$$\begin{aligned}\mathbf{y}_{u,k} =&\mathbf{h}_{u,k}\mathbf{X}_k[m] + \mathbf{z}_{u,k}\\ =&p_m \sum_{\substack{\mathcal{B}\in\mathcal{T}[m]\\ \mathcal{B}\neq\mathcal{B}_\circ}} \sum_{V\in\mathcal{B}} \sum_{v\in V} \underbrace{\mathbf{h}_{u,k}\mathbf{h}_k^{\perp}[\mathcal{N}(\mathcal{U}[m],\mathcal{B},V,v)]}_{\mathsf{Term}_1} \mathbf{w}_{d_v,k}^{\mathcal{B}\setminus\{V\}}\\ &+ p_m \sum_{\substack{V\in\mathcal{B}_\circ\\ V\neq V_\circ}} \sum_{v\in V} \mathbf{h}_{u,k}\mathbf{h}_k^{\perp}[\mathcal{N}(\mathcal{U}[m],\mathcal{B}_\circ,V,v)] \underbrace{\mathbf{w}_{d_v,k}^{\mathcal{B}_\circ\setminus\{V\}}}_{\mathsf{Term}_2}\\ &+ p_m \sum_{\substack{v\in V_\circ\\ v\neq u}} \underbrace{\mathbf{h}_{u,k}\mathbf{h}_k^{\perp}[\mathcal{N}(\mathcal{U}[m],\mathcal{B}_\circ,V_\circ,v)]}_{\mathsf{Term}_3} \mathbf{w}_{d_v,k}^{\mathcal{B}_\circ\setminus\{V_\circ\}}\\ &+ p_m \mathbf{h}_{u,k}\mathbf{h}_k^{\perp}[\mathcal{N}(\mathcal{U}[m],\mathcal{B}_\circ,V_\circ,u)]\, \mathbf{w}_{d_v,k}^{\mathcal{B}_\circ\setminus\{V_\circ\}}\\ &+ \mathbf{z}_{u,k}. \end{aligned} \tag{13}$$

We note that every term in the first summation is zero, since $\mathbf{h}_{u,k}\mathbf{h}_k^{\perp}[\mathcal{N}(\mathcal{U}[m],\mathcal{B},V,v)] = 0$ for every $\mathcal{B} \neq \mathcal{B}_\circ$, and hence $\mathsf{Term}_1 = 0$. The same holds for the third summation, since $\mathbf{h}_{u,k}\mathbf{h}_k^{\perp}[\mathcal{N}(\mathcal{U}[m],\mathcal{B}_\circ,V_\circ,v)] = 0$ for $u \in V_\circ \setminus \{v\}$, i.e., for all other users within the same meta-user $V_\circ$, and thus, $\mathsf{Term}_3 = 0$. On the other hand, the codeword chunks appearing in the second summation are indexed by $\mathcal{B}_\circ \setminus \{V\}$ for some $V \neq V_\circ$. Since $u \in V_\circ \in \mathcal{B}_\circ \setminus \{V\}$, the file segment $W_{d_v}^{\mathcal{B}_\circ\setminus\{V\}}$ is cached at user u. That is, Term_2 in the second summation can be reconstructed from the cache content of user u, and the corresponding term can be subtracted from $\mathbf{y}_{u,k}$. Therefore, considering the zero-forcing and the suppression of interference using cache content, user u obtains $p_m\mathbf{h}_{u,k}\mathbf{h}_k^{\perp}[\mathcal{N}(\mathcal{U}[m],\mathcal{B}_\circ,V_\circ,u)]\,\mathbf{w}_{d_v,k}^{\mathcal{B}_\circ\setminus\{V_\circ\}} + \mathbf{z}_{u,k}$ for all $k \in [K]$, from which the file segment $W_{d_u}^{\mathcal{B}_\circ\setminus\{V_\circ\}}$ can be decoded.

To summarize the interference mitigation scheme, consider a time block m and an active user $u \in V_\circ \in \mathcal{B}_\circ \in \mathcal{T}[m]$. The file segment sent for this user in this time block is $W_{d_u}^{\mathcal{B}_\circ\setminus\{V_\circ\}}$. This signal will be

1) zero-forced at every active user $v \in V$ if $V \notin \mathcal{B}_\circ$;
2) cached-out at any user v whose meta-user V belongs to $\mathcal{B}_\circ$ but $V \neq V_\circ$;
3) zero-forced at all other users who belong to the same meta-user, i.e., $v \in V_\circ \setminus \{u\}$,
4) and decoded at user u.

Therefore, the interference caused by this message will be cancelled at all active users except the intended user u.

Example 1. *Fig. 1 presents a network with $U \geq 36$ single-antenna users and a transmitter with $L = 28$ transmit antennas. The overall cache size in the network is sufficient to store $M = 8$ copies of the entire data base distributedly across the users. Recall that the meta-user size κ should be a common divisor of M and L. Among possible choices $\kappa \in \{1, 2, 4\}$, we choose $\kappa = 4$. Therefore, we have $\mathbb{M} = M/\kappa = 2$ and $\mathbb{L} = L/\kappa = 7$. Each meta-user covers $\kappa = 4$ users, e.g., $V_1 = \{1, 2, 3, 4\}$, $V_2 = \{5, 6, 7, 8\}$, etc. We assume that the number of users U is sufficiently large and divisible by $\kappa = 4$. The number of meta-users will be $\mathbb{U} = U/\kappa$.*

During the placement phase, each file W_n will be partitioned into $\binom{\mathbb{U}}{\mathbb{M}} = \binom{\mathbb{U}}{2}$ file segment, and each segment is labeled by a subset $\mathbb{S} \subset [\mathbb{U}]$ of size $|\mathbb{S}| = 2$. Then, each user u in a meta-user V_i caches all the file segments whose index include i. For instance since user 9 belongs to meta user V_3, the cache Z_9 includes all file segments of the form $W_n^{\{3,j\}}$ for any $3 \neq j \in [\mathbb{U}]$.

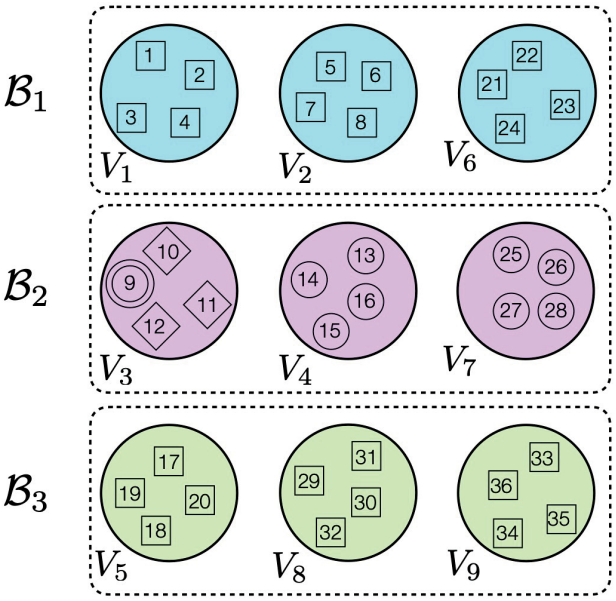

Fig. 1. Illustration of the meta-user and grouping being served in one time block for Example 1. Groups $\mathcal{B}_1$, $\mathcal{B}_2$, and $\mathcal{B}_3$ are served. Each group includes 3 meta-users, and each meta-user includes 4 users.

A valid scheduling $\mathcal{T}$ may include a time block m given by

$$\begin{aligned}\mathcal{T}[m] &= \{\mathcal{B}_1, \mathcal{B}_2, \mathcal{B}_3\}\\ &= \{\{1, 2, 6\}, \{3, 4, 7\}, \{5, 8, 9\}\}.\end{aligned}$$

In this time block the set of active users is given by

$$\mathcal{U}[m] = \bigcup\nolimits_{\mathcal{B}\in\mathcal{T}[m]} \bigcup\nolimits_{V\in\mathcal{B}} V = \bigcup\nolimits_{j=1}^{9} V_j = \{1, 2, \ldots, 36\}.$$

Let us focus on user 9 in the meta-user V_3 which appears in the group $\mathcal{B}_2$ (see Fig. 1). The file segment to be sent to this user in this time block is given by $W_{d_9}^{\mathcal{B}_2\setminus\{3\}} = W_{d_9}^{\{4,7\}}$. This file will be sent along a direction which is orthogonal to the channel of the users in

$$\mathcal{U}[m] \setminus (\cup_{V\in\mathcal{B}_2} V) \cup (V_3 \setminus \{9\})$$

where V_3 is the meta-user that includes user 9, and $\mathcal{B}_2$ is the group that includes V_3. Note that this set includes $6\kappa + \kappa - 1 = 27 = L - 1$ users and hence, zero-forcing is feasible using $L = 28$ antennas. These users are marked with either a square or a diamond in Fig. 1. Moreover, the users in set

$$V_4 \cup V_7 = \bigcup_{V\in\mathcal{B}_2, V\neq V_\circ} V = \{13, 14, 15, 16, 25, 26, 27, 28\}$$

belong to either V_4 or V_7, and hence, they have the file segment $W_{d_9}^{\{4,7\}}$ in their cache (these users are marked with a single circle in Fig. 1). Therefore, the interference caused by sending the message $W_{d_9}^{\{V_4,V_7\}}$ can be cancelled at all the active users in $\mathcal{U}[m]$, except the intended user $u = 9$.

Reciprocally, we can argue that when all users in $\mathcal{U}[m]$ are served, the codeword intended for users marked with squares or diamonds are zero-forced at user $u = 9$. Moreover, the codewords intended for users marked by circles are cached at user $u = 9$, and can be suppressed. Therefore, user $u = 9$ can decode its desired file segment. ◇

V. Scheme complexity

The subpacketization level, the number of packets that each file needs to be divided to, is a major source of complexity in cache-aided communication systems. The baseline for multi-antenna cache-aided communication is the scheme of Shariatpanahi et al. [2] (referred to as the SCK scheme). This scheme divides each file into $\binom{U}{M}$ segments in the placement phase, but further splits each packet at the delivery phase into $\binom{U-M-1}{L-1}$ sub-segments. Therefore, the overall subpacketization level is $\binom{U}{M}\binom{U-M-1}{L-1}$, which is prohibitively large.

The scheduling scheme avoids the second division, and hence has a subpacketization level of only $\binom{U}{M}$. The LE scheme reduces the subpacketization level even further, and requires only $\binom{U/L}{M/L}$ sub-packets. But, the LE scheme is feasible only if L divides M.

The scheme proposed in this paper takes the benefits of both the scheduling and the LE schemes, and has a subpacketization level of $\binom{U/\kappa}{M/\kappa}$, while it works for any integer κ that divides both M and L. Thus, it is always at least as good as both schemes, and is applicable to wider range of network parameters compared to the LE scheme.

Note that by choosing $\kappa = 1$ (which is always allowed) our scheme exactly recovers to the scheduling scheme. On the other hand, if $\gcd(M, L) = L$ (or equivalently, if L divides M), then by choosing $\kappa = L$ our scheme exactly reduces to the LE scheme.

To illustrate the complexity of the different schemes, Table I compares the subpacketization level of the SCK scheme, the scheduling scheme (marked as Sch), our novel scheme (marked as New), and the LE scheme. The subpacketization level is shown as the $\log_{10}$ of the number of file segments required by each scheme, for a network with $L = 4$ transmit antennas and $U = 32$ users. Note that the table includes scenarios in which $(M + L)/(M + \kappa)$ is not integer. The implementation of such non-integer cases requires an extension of the presented scheduling scheme, and is eliminated here due to page limit. This extension will be presented in the journal version of this work.

The table demonstrates the fact that the subpacketization level of the proposed scheme never exceeds those of the scheduling scheme nor the LE scheme. In particular, for $M = 1, 3, 5, 7$ we have to use $\kappa = \gcd(M, L) = 1$. Thus, our scheme recovers the scheduling scheme and shows a similar complexity. On the other hand, for $M = 4, 8$ we can use $\kappa = L = 4$ and hence our scheme reduces to the LE scheme and shows a similar complexity. For other values of M (i.e., 2 and 6) our scheme has lower complexity than the scheduling scheme, while the LE scheme is not feasible.

More importantly, our scheme shows significant reduction in the subpacketization level compared to the SCK scheme for any parameter value. For the values presented in the table the complexity reduction varies from 3 to 8 orders of magnitude. This is a significant advantage, that offers a complexity that is feasible in practical networks.

M	$\log_{10}$(sub-packets)			
	SCK	Sch	New	LE
1	5.1	1.5	1.5	-
2	6.3	2.7	1.2	-
3	7.2	3.7	3.7	-
4	8	4.6	0.9	0.9
5	8.7	5.3	5.3	-
6	9.3	6	2.7	-
7	9.8	6.5	6.5	-
8	10	7	1.4	1.4

TABLE I
A comparison of the subpacketization level of the SCK scheme, the scheduling scheme (marked Sch) our novel scheme (marked New) and the LE scheme.

VI. Summary

This paper introduced a novel scheme for cache-aided communication with low complexity. The proposed scheme combines the benefits of the LE scheme (of Lampiris and Elia) and the scheduling scheme (of Mohajer and Bergel). The scheme guarantees the optimal DoF while its subpacketization level is always lower than or equal to the subpacketization level of the other schemes in the literature. The low complexity of the new scheme allows practical implementation of cache-aided communication even in networks with tens of users.

References

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

[2] S. P. Shariatpanahi, G. Caire, and B. H. Khalaj, "Multi-antenna coded caching," in *2017 IEEE International Symposium on Information Theory (ISIT)*. IEEE, 2017, pp. 2113–2117.

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

[4] S. Jin, Y. Cui, H. Liu, and G. Caire, "Uncoded placement optimization for coded delivery," in *2018 16th International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt)*. IEEE, 2018, pp. 1–8.

[5] K. Shanmugam, A. M. Tulino, and A. G. Dimakis, "Coded caching with linear subpacketization is possible using ruzsa-szeméredi graphs," in *IEEE ISIT*, 2017, pp. 1237–1241.

[6] L. Tang and A. Ramamoorthy, "Low subpacketization schemes for coded caching," in *2017 IEEE International Symposium on Information Theory (ISIT)*. IEEE, 2017, pp. 2790–2794.

[7] E. Lampiris and P. Elia, "Adding transmitters dramatically boosts coded-caching gains for finite file sizes," *IEEE Journal on Selected Areas in Communications*, vol. 36, no. 6, pp. 1176–1188, 2018.

[8] S. Mohajer and I. Bergel, "Miso cache-aided communication with reduced subpacketization," in *ICC 2020-2020 IEEE International Conference on Communications (ICC)*, 2020.

[9] I. Bergel and S. Mohajer, "Practical scheme for miso cache-aided communication," in *2020 IEEE 21st International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*. IEEE, 2020, pp. 1–5.

[10] S. Mohajer and I. Bergel, "Reduced subpacketization in miso cachingvia user scheduling," in *submitted*, 2021.

ISBN 978-3-8007-5686-5