Power Efficient MISO Caching With Practical Subpacketization via User Scheduling

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

***Abstract*— We present a novel power-efficient and low-complexity scheme for cache-aided communication in networks with a multi-antenna base station that serves multiple single-antenna users. The scheme is based on transmitting coded messages to disjoint groups of users simultaneously and achieves an important trade-off between performance and complexity. The subpacketization level of the proposed scheme is sub-optimum compared to the state-of-the-art but is still feasible for a practical range of network parameters. On the other hand, the scheme achieves near-optimal performance and asymptotically achieves the same degrees of freedom (DoF) as the best-known schemes achieve. However, compared to other optimum achievable rates, the proposed scheme suffers from minor performance degradation due to power loss, which becomes negligible as the signal-to-noise ratio or the number of users grows. In return, the reductions in complexity and subpacketization allow for practical implementation of this scheme even for a large number of users. The presented scheme is also very flexible to the variation of the network topology and can easily be generalized to heterogeneous and dynamic scenarios.**

***Index Terms*— Cache-aided communication, MISO, power efficiency, subpacketization level.**

I. Introduction

DESPITE recent improvements in wireless communication technologies and data delivery networks, the rates supported by these networks are not likely to keep up with the overwhelming growth in demand. Caching the data at the users is a strategy that offers benefits from off-peak hours and shifts a part of the traffic to lower network traffic time. The gain of traditional caching is limited to the cache size of each individual user, which is typically negligible in practice. The *cache-aided communication* technique was introduced by Maddah-Ali and Niesen in [3], which allows for a *global gain*, in addition to the gain offered by traditional caching. The global gain is due to the possibility of interference cancellation at user v if a message sent to user u is cached by user v. This yields a broadcasting opportunity that serves multiple users simultaneously. The global gain scales with the aggregate size of the cache distributed among all the users in the network and can be substantial due to the large number of users in the network.

The *cache-aided communication* or *coded caching* scheme consists of a placement phase and a delivery phase [3]. During the placement phase, before knowing the users' requests, we can store some packets from the database in each user's memory. Once the requests are revealed, the server generates a set of coded messages and transmits them to the users 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 utility of a packet cached at one user, even if it is requested by another user. This provides an opportunity for *multicasting combined packets* intended for many users and increases the number of achievable degrees of freedom (DoF) by the number of copies of the database cached among all the users.

The application of coded caching in wireless communication has received significant attention in recent years. In particular, [4], [5], [6], [7], [8], [9] studied coded caching in wireless networks in the presence of fading and/or erasure channels. Coded caching in wireless networks with multiple antennas at transmitters and/or receivers are considered in [6], [10], [11], and [12]. Employment of coded caching in wireless networks, and in particular, in cellular networks, requires addressing several practical issues [13]. In a practical system, each user has a channel with different statistics and capacities. Cache allocation should be optimized depending on network traffic, the user's channel quality, the user's available storage, and other network characteristics. Coded caching for heterogeneous networks with different channels and rates for users (in the delivery phase) is studied in [14] for networks with single transmit and receive antennas. In [14], each packet transmission is subject to the rate of the weakest user among those supposed to decode the packet.

A transmitter with L transmit antennas can spatially multiplex its data to L users and achieve a DoF of L. A homogeneous (with statistically identical channel links) MISO network is studied in [12], where the spatial multiplexing and multicasting gains are combined, and it is shown that the performance improves as the number of users grows. Interestingly, the spatial diversity gain and caching

Manuscript received 28 December 2022; revised 29 June 2023 and 28 September 2023; accepted 29 October 2023. Date of publication 21 November 2023; date of current version 12 June 2024. The work of Soheil Mohajer was supported in part by the National Science Foundation under Grant CCF-1749981. An earlier version of this paper was presented in part at the 2020 IEEE International Conference on Communications (ICC) [1] and in part at the 2020 IEEE International Workshop on Signal Processing Advances in Wireless Communications (SPAWC) 2020 [2]. The associate editor coordinating the review of this article and approving it for publication was Y. Cui. *(Corresponding author: Itsik Bergel.)*

Soheil Mohajer is with the Department of Electrical and Computer Engineering, University of Minnesota, Minneapolis, MN 55455 USA (e-mail: soheil@umn.edu).

Itsik Bergel is with the Faculty of Engineering, Bar-Ilan University, Ramat Gan 52900, Israel (e-mail: itsik.bergel@biu.ac.il).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TWC.2023.3332417.

Digital Object Identifier 10.1109/TWC.2023.3332417

gains can be simultaneously achieved. In [11], Shariatpanahi, Caire, and Khalaj showed that $L+M$ degrees of freedom could be achieved in a broadcast system with L transmit antennas at the server and an aggregate cache size that can distributedly store M copies of the database across the users. Throughout this paper, we will refer to the scheme of [11] as the SCK scheme. It is shown in [15] that $L+M$ is the maximum DoF that can be achieved with *uncoded cache placement*, i.e., the pre-fetched data is a subset of raw packets of the files in the database, and *one-shot linear data delivery*, in which no data is transmitted more than once.

A critical concern in adopting cache-aided communication for practical systems is the *subpacketization level*, which refers to the number of segments to which each file must be divided. A large subpacketization level leads to a complex and computationally heavy scheme, where many short-length file segments must be individually encoded at the transmitter and decoded by the users. The problem of subpacketization in cache-aided communication was widely studied for the single antenna setting [16], [17], [18], [19]. In particular, the problem is formulated as an optimization problem in [17] and [20]. The tradeoff between the delivery rate and the subpacketization is studied in [21], where some classes of codes with the optimum tradeoff are introduced. Moreover, for a specific range of parameters, combinatorial solutions are proposed [22], [23].

For the MISO setting with U users, the SCK scheme [11] requires dividing each file into $\binom{U}{M}\binom{U-M-1}{L-1} = O\left(U^{M+L-1}\right)$ file segments. This is practically infeasible, especially since cache-aided communication is attractive mostly for networks with a large number of users. A placement strategy based on hypercube combinatorial design is proposed in [24] for an interference channel with multiple transmitters and receivers. Applying this scheme to a MISO network, we get a subpacketization level of $\left(\frac{U}{M}\right)^M \binom{U/M-2}{L/M-1}\binom{U/M-1}{L/M}^{M-1}\left(\left(\frac{L}{M}\right)!\right)^M \frac{M!}{L} = O\left(U^{M+L-1}\right)$, which is the same order as that of the SCK scheme. Note that in both schemes [11], [24], the exponent of subpacketization not only increases with the cache size but also with the number of antennas.

In a seminal work, Lampiris and Elia [25] proposed a scheme based on grouping and cache replication ideas. The scheme of [25] treats the network as if there are only U/L *effective* users, yet it achieves the optimum DoF of $M+L$. The main drawback of this scheme is the requirement for the number of copies of the database distributedly cached across the users to be divisible by the number of antennas, i.e., M/L should be an integer. In practice, M is typically small, and hence, such a constraint is often not feasible. If this requirement is not satisfied, the scheme's efficiency decreases, and the achievable DoF reduces by a multiplicative factor of at most 2, compared to the optimal DoF. Even when $L|M$, and the scheme achieves the optimal DoF, it still suffers from a significant power loss compared to the SCK scheme [11], due to the replacement of XOR operations with arithmetic signal additions in the real or complex field.

In a recent pioneering paper, Salehi et al. introduced a cyclic caching for the regime of $L \geq M$, for which the subpacketization level is $(M+L)U$, which only scales linearly with the number of users [26]. This subpacketization level is very convenient, and the applicability to $L \geq M$ covers a range of parameters that are not covered in [25]. But, this scheme suffers from the same power loss as [25] due to streaming one message for each active user, compared to the proposed scheme in which we send coded messages. As we will discuss later, this power loss can be as large a multiplicative factor of $M+1$, which is significant in some scenarios.

In this work, we present a low-complexity cache-aided communication scheme for multiple-antennas systems, which is applicable to any integer value of M. Our proposed scheme requires a subpacketization level of only $\binom{U}{M} = O\left(U^M\right)$, which is significantly smaller than $O\left(U^{M+L-1}\right)$ required by the SCK scheme. Yet, we prove that the scheme performs nearly as well as the SCK scheme and significantly better than [25] and [26] in terms of power efficiency. Moreover, unlike the SCK scheme, where users are simultaneously decoding multiple messages, in the proposed scheme, each active user receives and decodes only one message at each transmission slot. While the scheme is presented for a homogeneous network where users experience fading channels with identical statistics, it can be extended to arbitrary network topologies (see, for example [27].)

Paper Organization: In the following, we first present the system model in Section II. The proposed scheme is described in Section III. Our scheme requires scheduling, for which three algorithms are discussed in Section IV. Section V presents an extensive analytical comparison between the proposed and the SCK schemes. Our analytical comparisons are corroborated by numerical simulation presented in Section VI, and we conclude the paper in Section VII.

Notation: For an integer n we use $[n]$ to refer to set $\{1, 2, \ldots, n\}$. For 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 and calligraphic letters (e.g. $\mathcal{A}$) to refer to sets. The expected value is denoted by $\mathbb{E}[\cdot]$, the binary XOR operation is represented by $\oplus$, and $\lfloor x \rfloor$ and $\lceil x \rceil$ denote the floor and ceiling of x, respectively.

II. System Model

We consider a network with U users, each with one receive antenna and a single base station (BS) equipped with L transmit antennas. We focus on a wideband 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 nth received sample after matched filtering for the kth frequency bin at user u is given by

$$rCly_{u,k}[n] = \mathbf{h}_{u,k}\mathbf{x}_k[n] + z_{u,k}[n], \tag{1}$$

where $\mathbf{x}_k[n] \in \mathbb{C}^{L\times 1}$ is the transmit vector at time n, $z_{u,k}[n] \sim \mathbb{CN}\left(0, \frac{B}{K}\right)$ is the additive white Gaussian noise sample at User u in the frequency bin k, and $\mathbf{h}_{u,k} \in \mathbb{C}^{1\times L}$ 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 $P = \mathbb{E}\left[\|\mathbf{x}_k[n]\|^2\right]$. We also assume a *homogeneous* network, where the fading channels are distributed as $\mathbf{h}_{u,k} \sim \mathbb{CN}(\mathbf{0}, \mathbf{I})$ for every user $u \in [U]$ and frequency band $k \in [K]$.

Finally, we assume that the perfect channel state information is available at the BS and the receivers.

The BS has a dictionary of N files, namely $\{W_1, W_2, \dots, W_N\}$, each of size F bits. Each user is interested in one of the files. 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. That is, 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 a file W_{d_u} from the dictionary, and the BS serves 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 subset of users $\mathcal{A} \subseteq [U]$ with $|\mathcal{A}| \le L-1$, 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 $\mathcal{A}$ at frequency bin k. That is, $\|\mathbf{h}_k^{\perp}[\![\mathcal{A}]\!]\| = 1$ and $\mathbf{h}_{u,k}\mathbf{h}_k^{\perp}[\![\mathcal{A}]\!] = \mathbf{0}$, for every $u \in \mathcal{A}$ and $k \in [K]$. 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 these conditions.

III. The Proposed Scheme

We use the placement strategy proposed in [3]. First, each file is split into $\binom{U}{M}$ segments, and labeled with subsets of $[U]$ of size M, namely, $W_n = \{W_n^{\mathcal{S}} : \mathcal{S} \subseteq [U], |\mathcal{S}| = M\}$ for every $n \in [N]$. Then, the cache of user $u \in [U]$ will be filled by all file segments with labels including u, i.e.,

$$Z_u = \bigcup\nolimits_{n\in[N]} \{W_n^{\mathcal{S}} : u \in \mathcal{S}\}. \tag{2}$$

This implies $|Z_u| = N\binom{U-1}{M-1}F/\binom{U}{M} = NMF/U$, and the cache size constraint is satisfied.

After the placement phase is completed, the users reveal their requests. Let W_{d_u} be the file requested by user u. The delivery phase is divided into several time slots. In each time slot, we can serve *up to* $M+L$ users, which are partitioned into *groups* of size $M+1$. Therefore, the number of groups served in each time slot is $g \triangleq \frac{M+L}{M+1}$. In the following, we first describe the delivery phase for integer values of g. Then, in Section III-B, we relax this constraint.

A. Integer Values of g

Consider all subsets of $[U]$ of size $M+1$, that is, $\mathcal{G} \triangleq \{\mathcal{B} \subseteq [U] : |\mathcal{B}| = M+1\}$. In the following, we refer to each such subset as a *group*. In each time slot m, we can serve g groups. In the following, we specify the properties of groups that can be served together.

Definition 1 (Scheduling for Integer g): For integer value of g, a scheduling *is a table $\mathcal{T}$ with T rows and g columns, where each row $\mathcal{T}[m]$ is a collection of at most g groups from $\mathcal{G}$, i.e., $\mathcal{T}[m] \subset \mathcal{G}$ and $|\mathcal{T}[m]| \le g$. Such scheduling is called valid if and only if*

(i) All groups are covered by $\mathcal{T}$, i.e., $\bigcup_{m=1}^{T} \mathcal{T}[m] = \mathcal{G}$.

(ii) All groups in each row 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 slot $m \in \{1, 2, \dots, T\}$, the row $\mathcal{T}[m]$ determines the set of groups to be served in time slot m. More precisely, the set of active users in time slot m is given by $\mathcal{U}[m] = \bigcup_{\mathcal{B}\in\mathcal{T}[m]} \mathcal{B}$. Each user $v \in \mathcal{B} \in \mathcal{T}[m]$ will be served by the file segment $W_{d_v}^{\mathcal{B}\setminus\{v\}}$ during the time slot m. To this end, we first form a coded file segment for the group $\mathcal{B}$, given by $W_{\mathcal{B}} \triangleq \bigoplus_{u\in\mathcal{B}} W_{d_u}^{\mathcal{B}\setminus\{u\}}$. Then, we modulate this coded file segment to a codeword $\mathbf{w}_{\mathcal{B}}$ of length τK. We further split the codeword into K chunks, namely $\mathbf{w}_{\mathcal{B},k} \in \mathbb{C}^{1\times\tau}$ for $k \in [K]$, and send each chunk in one frequency bin. The transmit signal for time slot m in frequency bin k will be formed by

$$\mathbf{X}_k[m] = \sqrt{P/K\cdot|\mathcal{T}[m]|} \sum_{\mathcal{B}\in\mathcal{T}[m]} \mathbf{h}_k^{\perp}[\![\mathcal{U}[m]\setminus\mathcal{B}]\!]\mathbf{w}_{\mathcal{B},k}, \tag{3}$$

where[1] $\mathbf{X}_k[m] \in \mathbb{C}^{L\times\tau}$ and its ℓ-th row will be sent over the ℓ-th transmit antenna at time slot m (which consists of τ clock cycles). Note that $|\mathcal{U}[n]\setminus\mathcal{B}| \le (g-1)(M+1) = \frac{L-1}{M+1}(M+1) = L-1$, and hence zero-forcing at users in $\mathcal{U}[n]\setminus\mathcal{B}$ is feasible using L transmit antennas.

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

$$\begin{aligned}
&\mathbf{y}_{u,k}[m] \\
&= \mathbf{h}_{u,k}\mathbf{X}_k[m] + \mathbf{z}_{u,k}[m] \\
&= \sqrt{\frac{P}{K\cdot|\mathcal{T}[m]|}} \sum_{\mathcal{B}\in\mathcal{T}[m]} \mathbf{h}_{u,k}\mathbf{h}_k^{\perp}[\![\mathcal{U}[m]\setminus\mathcal{B}]\!]\mathbf{w}_{\mathcal{B},k} + \mathbf{z}_{u,k}[m] \\
&\overset{(a)}{=} \sqrt{P/K\cdot|\mathcal{T}[m]|}\mathbf{h}_{u,k}\mathbf{h}_k^{\perp}[\![\mathcal{U}[m]\setminus\mathcal{B}_\circ]\!]\mathbf{w}_{\mathcal{B}_\circ,k} + \mathbf{z}_{u,k}[m],
\end{aligned} \tag{4}$$

where (a) is due to the fact that $\mathbf{h}_{u,k}$ is orthogonal to $\mathbf{h}_k^{\perp}[\![\mathcal{U}[m]\setminus\mathcal{B}]\!]$ for every $\mathcal{B} \ne \mathcal{B}_\circ$. Receiving $\{\mathbf{y}_{u,k}[m]\}$ for all $k \in [K]$, user u then decodes a single (coded) message $\mathbf{w}_{\mathcal{B}_\circ}$. Note that since $u \in \mathcal{B}_\circ$, all file segments $W_n^{\mathcal{B}_\circ\setminus\{v\}}$ (with $v \ne u$) are stored in the cache of user u, and they can be subtracted (XORed) from $\mathbf{w}_{\mathcal{B}_\circ}$ to recover the desired file segment $W_{d_u}^{\mathcal{B}_\circ\setminus\{u\}}$.

After the completion of the transmission scheme for all time slots $m \in [T]$, each user u will retrieve all its file segments $W_{d_u}^{\mathcal{S}}$ which are not cached in its memory, i.e., $u \notin \mathcal{S}$. Concatenating these segments with those in the cache, user u will be able to decode its desired file. The overall delay of service for all users in T time slots, each of duration τ, will be $T\tau$.

The following example demonstrates the delivery scheme discussed above.

Example 1: Consider the wireless network with $U = 8$ users that are served by a base station equipped with $L = 3$ antennas. Each user has a memory in which it can cache $M/U = 1/8$ of each file. Note that we have $g = \frac{L+M}{M+1} = \frac{4}{2} = 2$, which is an integer. In the placement phase we divide each file W_n into $\binom{U}{M} = 8$ equal size segments, and label them as $W_n^{\{1\}}, W_n^{\{2\}}, \dots, W_n^{\{8\}}$. Then user $u \in \{1, \dots, 8\}$ will cache all the file segments whose

[1]It is worth mentioning that achieving the optimal performance would require optimization of the power allocated to each frequency bin and each codeword chunk, subject to the BS average power constraint. Yet, for simplicity, we assume the power is equally distributed across the frequency bins and the data streams.

index is equal to $\{u\}$, that is, $Z_u = \{W_n^{\{u\}} : n = 1, \dots, N\}$. Without loss of generality, we assume that user u requests file W_u, i.e., $d_u = u$ for $u = 1, \dots, 8$. In the delivery phase, we need to serve $\binom{U}{M+1} = \binom{8}{2} = 28$ groups (pairs) of users. In each time slot, we can serve $g = 2$ groups; hence, we need $T = 28/2 = 14$ slots to complete the communication. To this end, we use the schedule $\mathcal{T}$ given in (5), as shown at the bottom of the page, where $(\cdot)^\intercal$ denotes the matrix transpose. Consider a time slot, say $m = 3$, with $\mathcal{T}[3] = \{\{1,4\},\{7,8\}\}$ during which the set of active users is $\mathcal{U}[3] = \{1,4,7,8\}$. The file segments to be sent during this time slot are $W_1^{\{4\}}$, $W_4^{\{1\}}$, $W_7^{\{8\}}$ and $W_8^{\{7\}}$, which form coded file segments $W_{\{1,4\}} = W_1^{\{4\}} \oplus W_4^{\{1\}}$ and $W_{\{7,8\}} = W_7^{\{8\}} \oplus W_8^{\{7\}}$. After modulation to codewords and computing the codeword chunks for each frequency bin, the base station sends

$$\mathbf{X}_k[3] = \sqrt{P/2K}\left(\mathbf{h}_k^\perp[\![\{7,8\}]\!]\mathbf{w}_{\{1,4\},k} + \mathbf{h}_k^\perp[\![\{1,4\}]\!]\mathbf{w}_{\{7,8\},k}\right),$$

in the frequency bin k. Let us consider user $u = 4$. Its received signal will be

$$\begin{aligned}\mathbf{y}_{4,k}[3] &= \mathbf{h}_{4,k}\mathbf{X}_{4,k}[3] + \mathbf{z}_{4,k}[3]\\ &= \sqrt{P/2K}\mathbf{h}_{4,k}\mathbf{h}_k^\perp[\![\{7,8\}]\!]\mathbf{w}_{\{1,4\},k} + \mathbf{z}_{4,k}[3], \quad (6)\end{aligned}$$

since $\mathbf{h}_{4,k}\mathbf{h}_k^\perp[\![\{1,4\}]\!] = \mathbf{0}$. Next, user 4 decodes $W_{\{1,4\}}$ from $\mathbf{y}_{4,k}[3]$ for all frequency bins $k \in [K]$. Recall that $W_1^{\{4\}} \in Z_4$, and hence, user 4 can recover $W_4^{\{1\}}$ by removing the interference, i.e., $W_4^{\{1\}} = W_{\{1,4\}} \oplus W_1^{\{4\}}$. Finally, user 4 can decode W_4 by concatenating the delivered and cached segments. All other users will be served in a similar manner in $T = 14$ time slots. $\diamond$

B. Non-Integer Values of g

When $g = \frac{M+L}{M+1}$ is not an integer, we can still serve $M+L$ users simultaneously. However, the scheduling becomes more complicated. In this case, the BS serves $\lfloor g \rfloor$ groups as before. In order to maintain the optimum DoF, it serves an additional $M + L - (M+1)\lfloor g \rfloor$ users. These additional users will be chosen from one or two other groups, called *partially served groups*. The remaining users of a partially served group must be served in different time slots. In the following, we present the properties of a valid scheduling for an arbitrary value of g.

Definition 2 (Scheduling for Arbitrary g): An $(M+1, M+L)$-scheduling $\mathcal{T}$ over $[U]$ is a table $\mathcal{T} = (\mathcal{T}[1], \mathcal{T}[2], \dots, \mathcal{T}[T])$, where each $\mathcal{T}[m]$ is a collection of pairs $(\mathcal{A},\mathcal{B})$, such that $\mathcal{B} \subseteq [U]$ is a subset of users of size $|\mathcal{B}| = M+1$ and $\mathcal{A} \subseteq \mathcal{B}$. A scheduling $\mathcal{T}$ is called valid *if it satisfies the following conditions*:

(C1) *All users in each group are served, i.e., for any $\mathcal{B} \in \mathcal{G}$, we have $\mathcal{B} = \bigcup_{m=1}^{T} \bigcup_{(\mathcal{A},\mathcal{B})\in\mathcal{T}[m]} \mathcal{A}$.*

(C2) *The set of active sub-groups in each time slot are disjoint, that is, $\mathcal{A} \cap \mathcal{A}' = \varnothing$, for every distinct pairs $(\mathcal{A},\mathcal{B})$ and $(\mathcal{A}',\mathcal{B}')$ which are served in the same time slot.*

(C3) *The interference caused by the message intended for other groups can be zero forced, i.e., for each time slot m and any $(\mathcal{A},\mathcal{B}) \in \mathcal{T}[m]$, we have $|\mathcal{U}[m]\backslash\mathcal{B}| \le L-1$, where $\mathcal{U}[m] \triangleq \bigcup_{(\mathcal{A},\mathcal{B})\in\mathcal{T}[m]} \mathcal{A}$.*

A pair $(\mathcal{A},\mathcal{B}) \in \mathcal{T}[m]$ indicates that each user $u \in \mathcal{A}$ will be served by a file segment $W_{d_u}^{\mathcal{B}\backslash\{u\}}$ in time slot m. When $\mathcal{A} = \mathcal{B}$, the group $\mathcal{B}$ is *completely* served. If $\mathcal{A} \subsetneq \mathcal{B}$, then $\mathcal{B}$ is a *partially* served group, i.e., only users in $\mathcal{A}$ are served, and the users in $\mathcal{B} \setminus \mathcal{A}$ still need to be served in other time slots. We illustrate a valid scheduling for a non-integer g in Example 2.

Remark 1: Note that the scheduling in Definition 1 for integer g is a special case of Definition 2, where all groups are completely served, i.e., $\mathcal{A} = \mathcal{B}$ for every $(\mathcal{A},\mathcal{B}) \in \mathcal{T}[m]$ and every $m \in [T]$.

Remark 2: If $(\mathcal{A},\mathcal{B}) \in \mathcal{T}[m]$ is partially served at time m, then Condition **(C1)** *implies that for each non-served user $v \in \mathcal{B}\backslash\mathcal{A}$ there exist some m' that v is served with file segment $W_{\mathcal{B}}$ in time slot m'. More precisely, there exist some m' and $\mathcal{A}'$ such that $v \in \mathcal{A}' \subset \mathcal{B}$ and $(\mathcal{A}',\mathcal{B}) \in \mathcal{T}[m']$, i.e., v is served in time m'.*

Consider a scheduling $\mathcal{T}$, and some pair $(\mathcal{A},\mathcal{B}) \in \mathcal{T}[m]$. The BS first forms a coded segment

$$W_{(\mathcal{A},\mathcal{B})} \triangleq \bigoplus_{u\in\mathcal{A}} W_{d_u}^{\mathcal{B}\backslash\{u\}}. \quad (7)$$

The BS modulates this file segment to a codeword $\mathbf{w}_{(\mathcal{A},\mathcal{B})}$, which will be then divided into K chunks, each of length τ, denoted by $\mathbf{w}_{(\mathcal{A},\mathcal{B}),k}$ for $k \in [K]$. Then the BS broadcasts

$$\mathbf{X}_k[m] = \sqrt{\frac{P}{K\cdot|\mathcal{T}[m]|}}\sum\nolimits_{(\mathcal{A},\mathcal{B})\in\mathcal{T}[m]} \mathbf{h}_k^\perp[\![\mathcal{U}[m]\setminus\mathcal{B}]\!]\mathbf{w}_{(\mathcal{A},\mathcal{B}),k},$$

at time slot m and frequency bin k. Here, $\mathbf{X}_k[m] \in \mathbb{C}^{L\times\tau}$ determines the transmit signal for each transmit antenna at each time slot of the slot. Note that Condition **(C3)** guarantees that $|\mathcal{U}[m]\backslash\mathcal{B}| \le L-1$ and hence zero-forcing using L transmit antennas is feasible.

The received signal at an active user $u \in \mathcal{A}_\circ$ with $(\mathcal{A}_\circ,\mathcal{B}_\circ) \in \mathcal{T}[m]$ is given by

$$\begin{aligned}&\mathbf{y}_{u,k}[m]\\ &= \mathbf{h}_{u,k}\mathbf{X}_k[m] + \mathbf{z}_{u,k}[m]\\ &= \sqrt{\frac{P}{K|\mathcal{T}[m]|}}\sum_{(\mathcal{A},\mathcal{B})\in\mathcal{T}[m]} \mathbf{h}_{u,k}\mathbf{h}_k^\perp[\![\mathcal{U}[m]\setminus\mathcal{B}]\!]\mathbf{w}_{(\mathcal{A},\mathcal{B}),k} + \mathbf{z}_{u,k}[m].\end{aligned}$$

Consider the term in the summation above corresponding to a pair $(\mathcal{A},\mathcal{B})$. If $u \in \mathcal{U}[m] \setminus \mathcal{B}$ then $\mathbf{h}_{u,k}\mathbf{h}_k^\perp[\![\mathcal{U}[m]\setminus\mathcal{B}]\!] = \mathbf{0}$,

$$\begin{aligned}\mathcal{T} &= \begin{bmatrix} \mathcal{T}[1] & \mathcal{T}[2] & \mathcal{T}[3] & \mathcal{T}[4] & \mathcal{T}[5] & \mathcal{T}[6] & \mathcal{T}[7] & \mathcal{T}[8] & \mathcal{T}[9] & \mathcal{T}[10] & \mathcal{T}[11] & \mathcal{T}[12] & \mathcal{T}[13] & \mathcal{T}[14] \end{bmatrix}^\intercal\\ &= \left[\begin{array}{c|c|c|c|c|c|c|c|c|c|c|c|c|c} \{1,2\} & \{1,3\} & \{1,4\} & \{1,5\} & \{1,6\} & \{1,7\} & \{1,8\} & \{2,4\} & \{2,5\} & \{2,6\} & \{2,7\} & \{2,8\} & \{3,6\} & \{3,7\} \\ \{3,4\} & \{5,6\} & \{7,8\} & \{2,3\} & \{4,5\} & \{6,8\} & \{3,5\} & \{5,7\} & \{6,7\} & \{4,8\} & \{3,8\} & \{4,6\} & \{4,7\} & \{5,8\} \end{array}\right]^\intercal, \quad (5)\end{aligned}$$

i.e., the corresponding interference is zero-forced at user u. Otherwise, we have $u \in \mathcal{B}$. In this case, if $\mathcal{A} \neq \mathcal{A}_\circ$, then Condition **(C2)** implies $\mathcal{A} \cap \mathcal{A}_\circ = \varnothing$, and more importantly, $u \notin \mathcal{A}$. Therefore, u has all the components of the coded message $W_{(\mathcal{A},\mathcal{B})}$ in its cache (see (7)). Hence, it can construct the coded version of $W_{(\mathcal{A},\mathcal{B})}$ and subtract it from the received signal. The output after subtracting the cached interference is given by

$$\begin{aligned}\tilde{\mathbf{y}}_{u,k}[m] &= \mathbf{y}_{u,k}[m] - \sqrt{\frac{P}{K|\mathcal{T}[m]|}} \sum_{\substack{(\mathcal{A},\mathcal{B})\in\mathcal{T}[m] \\ \mathcal{B}\setminus\mathcal{A}\ni u}} \mathbf{h}_{u,k}\mathbf{h}_k^{\perp}[\![\mathcal{U}[m]\setminus\mathcal{B}]\!]\mathbf{w}_{(\mathcal{A},\mathcal{B}),k} \\ &= \sqrt{\frac{P}{K|\mathcal{T}[m]|}}\mathbf{h}_{u,k}\mathbf{h}_k^{\perp}[\![\mathcal{U}[m]\setminus\mathcal{B}_\circ]\!]\mathbf{w}_{(\mathcal{A}_\circ,\mathcal{B}_\circ),k} + \mathbf{z}_{u,k}[m]. \quad (8)\end{aligned}$$

Since $u \in \mathcal{A}_\circ$, all file segments $W_n^{\mathcal{B}_\circ\setminus\{v\}}$ (with $v \neq u$) are stored in the cache of user u, and they can be removed (XORed) from $\mathbf{w}_{(\mathcal{A}_\circ,\mathcal{B}_\circ)}$ to recover the desired file segment $W_{d_u}^{\mathcal{B}_\circ\setminus\{u\}}$. Decoding the desired file segments for all other users follows from a similar argument.

Remark 3: Note that **(C3)** *implies that the number of active users in each time slot satisfies*

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

More importantly, this condition has a special implication for the feasibility of interference cancellation at partially served groups. Let $(\mathcal{A},\mathcal{B})$ be a partially served group in time slot m, i.e., users in $\mathcal{A}$ receive $W_{(\mathcal{A},\mathcal{B})}$ during the mth time slot. The other active users in $\mathcal{U}[m] \setminus \mathcal{A}$ should be able to cancel the interference caused by $W_{(\mathcal{A},\mathcal{B})}$. While we can only zero-force this message at $L-1$ users, we may have $|\mathcal{U}[m] \setminus \mathcal{A}| > L-1$. Nevertheless, this interference can also be canceled using the cached content at any user $u \in \mathcal{B} \setminus \mathcal{A}$. To this end, we need $W_{(\mathcal{A},\mathcal{B})}$ to be cached at some active users in $\mathcal{U}[m] \setminus \mathcal{A}$. This implies that even though user $u \in \mathcal{B} \setminus \mathcal{A}$ does not receive $W_{d_u}^{\mathcal{B}\setminus\{u\}}$, but it is active, i.e., there exists another $(\mathcal{A}',\mathcal{B}') \in \mathcal{T}[m]$ where $u \in \mathcal{A}'$.

For instance, $\mathcal{T}[1] = ((\{3\},\{1,2,3\}),(\{1,2,4\},\{1,2,4\}))$ is valid row for a schedule with $(M,L) = (2,2)$ (see Example 2 in Section IV-B). Here, $\mathcal{U}[1] = \{1,2,3,4\}$, and group $\{1,2,4\}$ is completely served, while group $\{1,2,3\}$ is partially served (only user 3 receives message $W_{d_3}^{\{1,2\}}$). The interference caused by $W_{d_3}^{\{1,2\}}$ can be cached out at users 1 and 2, and needs to be zero-forced only at user 4. In general, Condition **(C3)** *guarantees that there is a sufficiently large overlap between $\mathcal{U}[m] \setminus \mathcal{A}$ and $\mathcal{B} \setminus \mathcal{A}$, so that users in this intersection can cache out the interference caused by $W_{(\mathcal{A},\mathcal{B})}$.*

C. Beamforming Optimization

For simplicity of presentation, the delivery scheme proposed in this section focuses on zero-forcing, which is more efficient at high SNR. Yet, one should note that higher performance can be obtained by optimizing the number of served users and their beamforming vectors.

The beam optimization problem is not studied herein as it is identical to the problem solved in [28], where for a given set of active users and corresponding coded messages, the beams are optimized to maximize the throughput (for the worst active user). As a reference, we will only demonstrate the performance of such optimization in Fig. 5, in Section VI.

It is important to note that the *user scheduling* approach proposed in this work can be directly combined with the *beam optimization* of [28]. More specifically, when g is an integer, for every time slot, once the groups to be served are identified, the beam optimization problem is identical to the minimization problem in (37) of [28]. Thus, the exact same optimization can be applied to each time slot. The case of non-integer g can also be solved with minor adaptations.

However, the differences between the scheduling approach in this work and that of [28] are worth noting. First, while the scheme of [28] allows for simultaneously sending β messages for each user (using a multiple access channel decoder), we limit each user to decode only one message at a time ($\beta = 1$). Second, while only an integer number of groups can be served in [28], the proposed scheduling can be applied to all system parameters. Third, the subpacketization level of [28] is $O(U^{M+L-1})$, but the proposed scheme requires subpacketization of $O(U^{M+1})$.

Lastly, due to the symmetry of the delivery scheme of [28] with respect to all users, it turns out that our proposed scheduling corresponds to a subset of time slots used in [28]. In that sense, for general a channel model (e.g., a narrowband scenario), our specific choice of groups to be simultaneously served may lead to a higher or lower rate than [28]. A channel-aware greedy scheduling is proposed in [27], which selects the groups to be jointly served in order to maximize overall performance. The beam optimization of [28] can also be adopted for the scheduling of [27] to achieve superior performance. However, in a homogeneous wideband regime (that is the focus of this work), the performance is averaged over many narrowband channels, and the rate of the proposed scheme (with optimized beams) is identical to that of [28].

IV. The Scheduling Algorithms

The success of the delivery scheme presented in Section III substantially depends on the scheduling, which governs the selection of users to be served in each time slot. As we have $\binom{U}{M+1}$ groups to be scheduled, and at most $M+L$ users can be served at each time slot, optimal scheduling takes $\tau = \left\lceil \frac{M+1}{M+L}\binom{U}{M+1} \right\rceil$ time slots. When $g = \frac{M+L}{M+1}$ is an integer, such scheduling exists and can be generated in a recursive manner, as discussed in Section IV-A. Moreover, we introduce a greedy scheduling algorithm in Section IV-B that can be used for both integer and non-integer values of g, which is shown to be efficient through simulations. The greedy algorithm is proved to converge to optimal performance as the number of users increases, at least when g is an integer.

A. An Optimum Scheduling for Integer-Valued g

Let $g = \frac{M+L}{M+1}$ be an integer. Recall from Definition 1 that an optimum scheduling is indeed a partitioning of all possible groups in $\mathcal{G} = \{\mathcal{B} \subseteq [U] : |\mathcal{B}| = M+1\}$ into $T = \left\lceil \frac{M+1}{M+L}\binom{U}{M+1}\right\rceil$ rows, such that the groups in each row are pairwise disjoint.

A simple and optimal scheduling algorithm for this regime can be developed based on independent sets in a graph. Consider a graph $K_U^{M+1} \triangleq (\mathcal{G}, \mathcal{E})$, where $\mathcal{E} = \{(\mathcal{B}, \mathcal{B}') : \mathcal{B}, \mathcal{B}' \in \mathcal{G}, \mathcal{B} \cap \mathcal{B}' \neq \emptyset\}$. In other words, in K_U^{M+1} each vertex corresponds to an $(M+1)$-group, and two vertices $\mathcal{B}$ and $\mathcal{B}'$ are connected via an edge if and only if they are not disjoint. Then, a valid schedule introduced in Definition 1 reduces to a partitioning of K_U^{M+1} into T *independent sets*, each of size at most g. Here, an independent set (of vertices) is a subset of the vertices in a graph, where no two of which are adjacent. The following theorem guarantees the existence of such partitioning.

Theorem 1 ([29]): Let $a_1, a_2, \dots, a_T$ be positive integers such that $\sum_{j=1}^{T} a_j = \binom{U}{M+1}$. Then, the vertices in K_U^{M+1} can be partitioned into $\mathcal{G} = \mathcal{G}_1 \cup \mathcal{G}_2 \cup \cdots \cup \mathcal{G}_T$ where $|\mathcal{G}_j| = a_j$ and $|\mathcal{N}(u, \mathcal{G}_j) - \mathcal{N}(v, \mathcal{G}_j)| \leq 1$ for every $u, v \in [U]$ and every $j \in [T]$. Here, $\mathcal{N}(u, \mathcal{G}_j) := \sum_{\mathcal{B} \in \mathcal{G}_j : u \in \mathcal{B}} 1$ is the number of appearances of u in the partition $\mathcal{G}_j$.

Proof: See Theorem 1 in [29]. ■

To get the desired scheduling, we can set $T = \left\lceil \frac{1}{g}\binom{U}{M+1}\right\rceil$, $a_1 = \cdots = a_{T-1} = g$, and $a_T = \binom{U}{M+1} - (T-1)g \leq g$. Then, the set of vertices in partition $\mathcal{G}_j$ is exactly the set of groups to be served in time slot j. Note that for $j = 1, \dots, T$, we have

$$\sum_{u \in [U]} \mathcal{N}(u, \mathcal{G}_j) = \sum_{u \in [U]} \sum_{\mathcal{B} \in \mathcal{G}_j : u \in \mathcal{B}} 1 = \sum_{\mathcal{B} \in \mathcal{G}_j} \sum_{u \in \mathcal{B}} 1 = \sum_{\mathcal{B} \in \mathcal{G}_j} |\mathcal{B}|$$
$$\leq g|\mathcal{B}| = \frac{M+L}{M+1}(M+1) = M+L \leq U.$$

Now, if there exists some $u \in [U]$ and $j \in [T]$ with $\mathcal{N}(u, \mathcal{G}_j) \geq 2$, then $\sum_{u \in [U]} \mathcal{N}(u, \mathcal{G}_j) \leq U$ guarantees that there should be another $v \in [U]$ that satisfies $\mathcal{N}(v, \mathcal{G}_j) = 0$. This leads to $|\mathcal{N}(u, \mathcal{G}_j) - \mathcal{N}(v, \mathcal{G}_j)| \geq 2$, which is in contradiction with the property guaranteed by Theorem 1. Therefore, we have $\mathcal{N}(u, \mathcal{G}_j) \leq 1$ for every $u \in [U]$. This implies that each user appears at most once in each partition $\mathcal{G}_j$, and hence, the groups in $\mathcal{G}_j$ are disjoint.

B. The Greedy Algorithm

The optimal schedule in Section IV-A is rather complicated and can be only applied for integer values of g. Thus, it is useful to have a universal but perhaps suboptimal scheduling algorithm that works for any g with a practical complexity. In the following, we present a greedy approach for group scheduling that is applicable to any system parameters. We do not present a general analytical study of the algorithm's performance. However, Theorem 2 shows that the algorithm performs close to optimum for a certain range of parameters and a large enough U. More importantly, our numerical results in Section VI show that the algorithm outperforms the bound in Theorem 2 for several tested values of (U, M, L), with integer and non-integer g, (see Fig. 2).

The greedy algorithm described in Algorithm 1 squeezes as many users as possible into each row of the schedule. Recall that we aim at serving all the groups in $\mathcal{G}$. In each row, we serve some of the groups *completely*, and possibly some groups will be *partially* served. If all the $M+1$ users in a group are served with their designated message, then the group is called to be completely served. Otherwise, if a subset of the users in a group is served, then the group is called *partially-served* group. Note that the remaining users of a partially served group need to be served in other time slots of the schedule.

Algorithm 1 The Greedy Algorithm

```
   Input: Parameters $M$, $L$, $U$.
   Output: Collision-Free Scheduling $\mathcal{T}$.
1  Initializiation:
2      Global: $m = 0$;
3      $\tilde{\mathcal{G}}$ = All Subsets of $[U]$ of Size $M+1$;
4      $\beta = (M+L) \mod (M+1)$;
5      Buffer = $\varnothing$; $\mathcal{T}[\,]$ = An empty array;
6  while $|\tilde{\mathcal{G}}| > 0$ do
7      $m = m+1$, $\mathcal{A}_\cup = \varnothing$, $\mathcal{B} = \{1\}$;
8      if $\beta > 0$ then
9          if Buffer = $\varnothing$ then
10             $\mathcal{B}_\circ = \mathtt{Grp}(\varnothing, 0, \varnothing)$;
11             Buffer = $\mathcal{B}_\circ$;
12         $\mathcal{A}_\circ$ = Buffer$[1 : \min\{|\text{Buffer}|, \beta\}]$ ;
13         $\mathcal{B} = \mathtt{Grp}(\mathcal{B}_\circ \backslash \mathcal{A}_\circ, |\mathcal{B}_\circ \backslash \mathcal{A}_\circ|, \mathcal{A}_\circ)$;
14         if $\mathcal{B} \neq \varnothing$ then
15             Buffer = Buffer $\setminus \mathcal{A}_\circ$;
16             $\mathcal{T}[m] = \{(\mathcal{A}_\circ, \mathcal{B}_\circ), (\mathcal{B}, \mathcal{B})\}$;
17             $\mathcal{A}_\cup = \mathcal{A}_\circ \cup \mathcal{B}$;
18         else
19             $\mathcal{T}[m] = \{(\text{Buffer}, \mathcal{B}_\circ)\}$;
20             $\mathcal{A}_\cup$ = Buffer;
21             Buffer = $\varnothing$;
22     while $M+L-|\mathcal{A}_\cup| \geq M+1$ and $\mathcal{B} \neq \varnothing$ do
23         $\mathcal{B} = \mathtt{Grp}(\varnothing, 0, \mathcal{A}_\cup)$;
24         $\mathcal{T}[m] = \mathcal{T}[m] \cup \{(\mathcal{B}, \mathcal{B})\}$;
25         $\mathcal{A}_\cup = \mathcal{A}_\cup \cup \mathcal{B}$;
26     if $M+L-|\mathcal{A}_\cup| > 0$ and $\mathcal{B} \neq \varnothing$ then
27         $\mathcal{B}_\circ = \mathtt{Grp}(\mathcal{A}_\cup, |\mathcal{A}_\cup| - L + 1, \varnothing)$;
28         $\mathcal{A}_\circ = \mathcal{B}_\circ \setminus \mathcal{A}_\cup$ ;
29         $\mathcal{T}[m] = \mathcal{T}[m] \cup \{(\mathcal{A}_\circ, \mathcal{B}_\circ)\}$;
30         Buffer = $\mathcal{B}_\circ \setminus \mathcal{A}_\circ$;
31 if |Buffer| $> 0$ then
32     $\mathcal{T}[m+1] = \{(\mathcal{A}_\circ, \mathcal{B}_\circ)\}$;
```

We serve at most $\lfloor g \rfloor$ complete groups in each time slot. Let $\beta = M+L-(M+1)\lfloor g \rfloor$. If $\beta = 0$, all groups can be served *completely*. If $\beta > 0$, the additional β users (from one or two other partially-served groups) will be served in the time slot.

The algorithm uses a Buffer that contains the users of a partial group that are yet to be served. This Buffer is initially empty, but once a partial group of size $M+1$ is selected, its users are added to Buffer. Then, they will be removed from

Algorithm 2 The Function Grp

```
1 Function $\mathcal{B} = \mathtt{Grp}(\mathcal{C}, m, \mathcal{A})$
2     $\mathcal{Q} = \{\mathcal{X} \in \tilde{\mathcal{G}} : |\mathcal{X} \cap \mathcal{C}| = m$ and $\mathcal{X} \cap \mathcal{A} = \varnothing\}$
3     $\mathcal{B} = \arg\max_{\mathcal{X} \in \mathcal{Q}} \mathsf{Scr}(\mathcal{X}, \tilde{\mathcal{G}})$;
4     $\tilde{\mathcal{G}} = \tilde{\mathcal{G}} \setminus \mathcal{B}$;
```

the Buffer as we serve them in some time slots. The selection of users from partial groups is as follows:

• if $\beta > |\text{Buffer}|$, serve β users from Buffer, and remove them from Buffer.

• otherwise, serve all users in Buffer. At the end of the scheduling, choose another partial group $\mathcal{B}_\circ$, and serve $\beta - |\text{Buffer}|$ users from $\mathcal{B}_\circ$. The remaining users of $\mathcal{B}_\circ$ will be added to Buffer.

• A maximum of $\lfloor g \rfloor$ complete groups that satisfy Definition 2 will be served in each time slot. The selection of complete groups is based on the following criteria:

– The first complete group includes the non-served users of the partial group, i.e., $\mathcal{B}_\circ \setminus \mathcal{A}_\circ$.

– They should be disjoint from the set of active users in other groups served in the time slot.

– Among all groups satisfying the above properties, the group with the maximum score is selected by Algorithm 2. Note that $\mathsf{Scr}(\mathcal{X}, \tilde{\mathcal{G}})$ is a function that prioritizes a group among all remaining groups. Intuitively, one would like to choose a group whose members have previously been minimally served. In particular, in our simulation results, we use

$$\mathsf{Scr}(\mathcal{X}, \tilde{\mathcal{G}}) = \sum\nolimits_{x \in \mathcal{X}} |\{\mathcal{Y} \in \tilde{\mathcal{G}} : x \in \mathcal{Y}\}|, \tag{9}$$

which can be efficiently computed and updated throughout the procedure. However, the algorithm works for a wide class of scoring functions.

• If the algorithm fails to find a group satisfying the above properties, it starts a new time slot.

An intuitive explanation for the success of the algorithm is as follows. When the number of users is large, there are many possible choices to fill the schedule of each time slot. Hence, the greedy algorithm will almost always succeed in allocating $M+L$ users in each slot. A conflict where no available groups comply with the conditions of Definition 2 may only occur at the end of the allocation process. This will yield serving less than $M+L$ in the last time slots. This leads to a reduction in the DoF. However, it turns out that the overall overhead normalized by the required number of rows of the schedule is negligible.

Example 2: Consider a wireless network with $U = 5$ users, each has a memory to cache $M/U = 2/5$ of each file. A base station with $L = 2$ antennas can serve (at most) $L + M = 4$ users in each time slot. In the placement phase we divide each file W_n into $\binom{U}{M} = 10$ equal size segments, and label them as $W_n^{\{1,2\}}, W_n^{\{1,3\}}, \dots, W_n^{\{4,5\}}$. Then user $u \in \{1, \dots, 5\}$ will cache all the file segments whose index is of the form $\{u, v\}$, for some $v \in \{1,2,3,4,5\} \setminus \{u\}$. Without loss of generality, we assume user u requests file W_u, i.e., $d_u = u$ for $u \in [5]$.

We serve groups of size $M + 1 = 3$. Each time slot can serve up to $\beta = M + L - (M+1)\lfloor g \rfloor = 1$ users from a partially served group. As shown in Table I, initially, all the groups are candidates to be partially served. The greedy algorithm starts by group $\{1,2,3\}$ and identifies its subset $\{1\}$ of size $\beta = 1$ to be partially served at time slot $m = 1$. For the complete group of the first time slot, the algorithm picks a group that (1) includes $\{2,3\}$, the non-served members of the partially-served group, and (2) has the lowest score among the candidates. As shown in the table, the candidates are $\{2,3,4\}$ and $\{2,3,5\}$, and both have score 2, so, w.o.l.g., $\{2,3,4\}$ will be selected. Hence, in this time slot, we have $\mathcal{T}[1] = \{(\{1\}, \{1,2,3\}), (\{2,3,4\}, \{2,3,4\})\}$. The server forms coded messages $W_{(\{1\},\{1,2,3\})} = W_1^{\{2,3\}}$ and $W_{(\{2,3,4\},\{2,3,4\})} = W_2^{\{3,4\}} \oplus W_3^{\{2,4\}} \oplus W_4^{\{2,3\}}$, modulates them, finds the codeword chunks for each frequency bin, and sends

$$\mathbf{X}_k[1] = \sqrt{\frac{P}{2K}} \Big(\mathbf{h}_k^{\perp} [\![\{1\}]\!] \mathbf{w}_{(\{2,3,4\},\{2,3,4\}),k} + \mathbf{h}_k^{\perp} [\![\{4\}]\!] \mathbf{w}_{(\{1\},\{1,2,3\}),k} \Big), \; k \in [K].$$

Note that this transmission serves all the active users in $\mathcal{U}[1] = \{1,2,3,4\}$, since users 1, 2, 3, and 4 can decode subfiles $W_1^{\{2,3\}}$, $W_2^{\{3,4\}}$, $W_3^{\{2,4\}}$, and $W_4^{\{2,3\}}$, respectively.

In the next time slot, we still have to serve $\{2\}$ from the partially-served group $\{1,2,3\}$. The server can choose between $\{1,3,4\}$ and $\{1,3,5\}$ for the complete group. However, since user 5 is never served, the hence, $\{1,3,5\}$ has a lower score and will be selected. Then, we have $\mathcal{T}[2] = \{(\{2\}, \{1,2,3\}), (\{1,3,5\}, \{1,3,5\})\}$.

In time slot $m = 3$, after serving the remaining part of $\{1,2,3\}$, i.e., $\{3\}$, the server can choose between $\{1,2,4\}$ and $\{1,2,5\}$, which have equal scores and hence $\{1,2,4\}$ is selected.

For $m = 4$, we need a new group to be partially served. Among the remaining groups, $\{1,4,5\}$, $\{2,4,5\}$, and $\{3,4,5\}$ have the minimum score, and hence $\{1,4,5\}$ is selected, from which user $\{1\}$ will be served. Our candidates for a complete group are $\{2,4,5\}$ and $\{3,4,5\}$, and the former is chosen as they have identical scores. Thus, $\mathcal{T}[4] = \{(\{1\}, \{1,4,5\}), (\{2,4,5\}, \{2,4,5\})\}$.

As can be seen in Table I, the overall schedule is complete in $T = 8$ time slots. Note that the minimum number of time slots is also $\left\lceil \frac{1}{g}\binom{U}{M+1} \right\rceil = \left\lceil \frac{3}{4} 10 \right\rceil = 8$, and hence the schedule is optimum. Nevertheless, in time slot $m = 8$, the schedule only serves 2 users instead of $M + L = 4$ users. This leads to a tiny DoF reduction. We discuss the overhead and gap to optimality of the proposed delivery method in Section V. ⋄

The following theorem proved in Appendix A provides an upper bound for the duration of the schedule obtained by the greedy algorithm.

Theorem 2: (a). For an integer g and any score function, the duration of the schedule offered by the greedy algorithm is upper-bounded by

$$T \leq \frac{1}{g}\binom{U}{M+1} + \frac{g-1}{g}\frac{L-1}{M!}U^M + 1. \tag{10}$$

TABLE I
THE EXECUTION OF ALGORITHM 1 FOR SYSTEM PARAMETERS $(U, M, L) = (5, 2, 2)$ AND $g = 4/3$, THAT TAKES $T = 8$ TIME SLOTS. A CIRCLE AROUND THEIR SCORE MARKS GROUP CANDIDATES, AND THE SELECTED GROUPS ARE SHOWN BY ⊠

m	$\mathcal{T}[m]$	User's Scr					Group's Score									
		1	2	3	4	5	1,2,3	1,2,4	1,2,5	1,3,4	1,3,5	1,4,5	2,3,4	2,3,5	2,4,5	3,4,5
0		0	0	0	0	0	⓪	⓪	⓪	⓪	⓪	⓪	⓪	⓪	⓪	⓪
1	$(\{1\}, \{1,2,3\})$	1	1	1	0	0	⊠	2	2	2	2	1	②	②	1	1
	$(\{2,3,4\}, \{2,3,4\})$	1	2	2	1	0		4	3	4	3	2	⊠	4	3	3
2	$(\{2\}, \{1,2,3\})$							4	3	④	③	2		4	3	3
	$(\{1,3,5\}, \{1,3,5\})$	2	2	3	1	1		5	5	6	⊠	4		6	4	5
3	$(\{3\}, \{1,2,3\})$							⑤	⑤	6		4		6	4	5
	$(\{1,2,4\}, \{1,2,4\})$	3	3	3	2	1		⊠	7	8		⑥		7	⑥	⑥
4	$(\{1\}, \{1,4,5\})$	4	3	3	3	2			9	10		⊠		8	⑧	⑧
	$(\{2,4,5\}, \{2,4,5\})$	4	4	3	4	3			11	11				10	⊠	10
5	$(\{4\}, \{1,4,5\})$								⑪	11				10		10
	$(\{1,2,5\}, \{1,2,5\})$	5	5	3	4	4			⊠	12				12		11
6	$(\{5\}, \{1,4,5\})$									⑫				12		11
	$(\{1,3,4\}, \{1,3,4\})$	6	5	4	5	4				⊠				⑬		⑬
7	$(\{2\}, \{2,3,5\})$	6	6	5	5	5								⊠		⑮
	$(\{3,4,5\}, \{3,4,5\})$	6	6	6	6	6										⊠
8	$(\{3,5\}, \{2,3,5\})$	6	6	6	6	6										

(b). For $L = 2$, there exists a score function such that the duration of the schedule offered by the greedy algorithm is upper-bounded by

$$T \leq \frac{1}{g}\binom{U}{M+1} + \frac{g-1}{g}(e-1)U^M + M + 1, \tag{11}$$

where $e \approx 2.718$ is the base of the natural logarithm.

We will use this theorem in Section V to theoretically bound the gap between the duration of the greedy-based schedule and that of an optimum schedule, where we show that the overhead of this (possibly sub-optimum) scheduling is of the order of $1/U$, and thus, is negligible for sufficiently large U. Our simulation results in Section VI suggest that the algorithm offers a vanishing overhead (as U grows) for general parameters, even though the theoretical proof for this claim is challenging. Moreover, even though the proof of part (b) is presented for a specific score function, in our simulations, we use the score function proposed in (9), as it is not only easier to evaluate but also demonstrates better overall performance.

Finally, it is worth emphasizing that the complexity of the greedy algorithm is due to the updating and minimizing the groups' scores (Line 3 of Algorithm 2), and it is in the order of $|\mathcal{G}|^2(M+1)$, which practical compared to the overall complexity of the network.

V. Performance Comparison

In this section, we analyze the performance of the proposed scheme and compare it with that of the SCK scheme [11]. The main factors of interest in our comparison are the receiver complexity, the subpacketization level, and the overall delivery delay. The comparison is also summarized and compared to other schemes in the literature in Table II below.

A. Receiver Complexity

In the SCK scheme, in each time slot, the BS simultaneously sends $\binom{M+L}{M+1}$ messages, out of which each active user is required to decode $\binom{M+L-1}{M}$ messages. Thus, each user should simultaneously decode all desired packets as in a multiple access channel (MAC). Such a receiver is quite complex and challenging to implement. In contrast, our scheme transmits only one packet to each active user at any given time. Thus, the receiver complexity is much lower.

B. The Subpacketization Level

Recall that for the scheme proposed in this paper, each file is split into $\binom{U}{M}$ segments. The delivery phase is solely based on combining packets of size $F/\binom{U}{M}$, encoding them using error correction codes, and modulating and sending them from the BS. For the SCK scheme, however, even though each file is initially divided into $\binom{U}{M}$ segments in the placement phase, further splitting is required for the delivery phase. More precisely, each segment is split into $\binom{U-M-1}{L-1}$ subsegments. Therefore, the overall subpacketization level is $\binom{U}{M}\binom{U-M-1}{L-1}$, which is substantially larger than that of our proposed scheme. Moreover, the overall number of codewords to broadcast by the proposed scheme is $\binom{U}{M+L}$, while this number is $\binom{U}{M+L}\binom{M+L}{M+1}$ for the SCK scheme.

C. The Overall Delay

In the following, we first characterize the duration (communication length) of the SCK scheme. Later, we use this as a baseline to evaluate the potential overhead of the proposed scheme compared to the SCK scheme.

Overall Delay of the SCK Scheme: The SCK scheme serves $M + L$ users in each time slot. Consider a time slot dedicated to users of set $\mathcal{Q}$ (with $|\mathcal{Q}| = M + L$). The BS sends a total of $\binom{M+L}{M+1}$ coded messages in each time slot, out of which $\binom{M+L-1}{M}$ messages are intended for each user $u \in \mathcal{Q}$. Thus, the *total* received power at user $u \in \mathcal{Q}$ in frequency bin k is

$$\begin{aligned}&\frac{P}{\binom{M+L}{M+1}K}\sum_{\substack{\mathcal{V}u\in\mathcal{V}\subseteq\mathcal{Q}\\|\mathcal{V}|=M+1}}\left|\mathbf{h}_{u,k}\mathbf{h}_k^{\perp}[\![\mathcal{Q}\setminus\mathcal{V}]\!]\right|^2\\&=\frac{\binom{M+L-1}{M}P}{\binom{M+L}{M+1}K}\frac{1}{\binom{M+L-1}{M}}\sum_{\substack{\mathcal{V}:u\in\mathcal{V}\subseteq\mathcal{Q}\\|\mathcal{V}|=M+1}}\eta_{u,k,\mathcal{Q}\setminus\mathcal{V}}\\&=\frac{P}{gK}\frac{1}{\binom{M+L-1}{M}}\sum_{\substack{\mathcal{V}:u\in\mathcal{V}\subseteq\mathcal{Q}\\|\mathcal{V}|=M+1}}\eta_{u,k,\mathcal{Q}\setminus\mathcal{V}},\end{aligned}$$

where $\eta_{u,k,\mathcal{D}} = \left|\mathbf{h}_{u,k}\mathbf{h}_k^{\perp}[\![\mathcal{D}]\!]\right|^2$. Since $\mathbf{h}_{u,k}$ has a Gaussian distribution and $\mathbf{h}_k^{\perp}[\![\mathcal{D}]\!]$ is unit-norm, the random variable $\eta_{u,k,\mathcal{D}}$ admits an exponential distribution with mean of 1. In a wideband communication scenario with bandwidth B, which is divided into K frequency bins, the maximal decodable sum-rate for messages intended for an active user $u \in \mathcal{Q}$ (in the MAC) is given by

$$\sum_{k=1}^{K}\frac{B}{K}\log_2\left(1+\frac{P}{gB}\frac{1}{\binom{M+L-1}{M}}\sum_{\substack{\mathcal{V}:u\in\mathcal{V}\subseteq\mathcal{Q}\\|\mathcal{V}|=M+1}}\eta_{u,k,\mathcal{Q}\setminus\mathcal{V}}\right). \quad (12)$$

Using the law of large numbers ($K \to \infty$) for the wideband regime, the rate of (12) converges to

$$R^{\mathsf{SCK}} = B\mathbb{E}\left[\log_2\left(1+\frac{P}{gB}\frac{1}{\binom{M+L-1}{M}}\sum_{\substack{\mathcal{V}:u\in\mathcal{V}\subseteq\mathcal{Q}\\|\mathcal{V}|=M+1}}\eta_{u,k,\mathcal{Q}\setminus\mathcal{V}}\right)\right]. \quad (13)$$

Note that this rate is the sum-rate of $\binom{M+L-1}{M}$ messages where each message is of length $F/\binom{U}{M}\binom{U-M-1}{L-1}$. Therefore, the minimum length of each time slot is given by

$$\tau^{\mathsf{SCK}} = \frac{F}{\binom{U}{M}\binom{U-M-1}{L-1}}\left(\frac{R^{\mathsf{SCK}}}{\binom{M+L-1}{M}}\right)^{-1} = \frac{\binom{M+L-1}{M}}{\binom{U}{M}\binom{U-M-1}{L-1}}\frac{F}{R^{\mathsf{SCK}}}.$$

Finally, note that the scheme requires a total of $T^{\mathsf{SCK}} = \binom{U}{M+L}$ time slots to serve all groups of size $|\mathcal{Q}| = M+L$. Therefore, the overall minimum delay of the SCK scheme is given by

$$D^{\mathsf{SCK}} = T^{\mathsf{SCK}}\tau^{\mathsf{SCK}} = \frac{U-M}{M+L}\frac{F}{R^{\mathsf{SCK}}}. \quad (14)$$

Delay Analysis of the Proposed Scheme: Let $\mathcal{T}$ be a scheduling satisfying **(C1)–(C3)** in Definition 2. The received power at an active user u in group $\mathcal{B}_\circ \in \mathcal{T}[m]$ is given by (see (8))

$$\frac{P}{K\cdot|\mathcal{T}[m]|}\eta_{u,k,\mathcal{U}[m]\setminus\mathcal{B}_\circ}, \quad (15)$$

where $\eta_{u,k,\mathcal{U}[m]\setminus\mathcal{B}_\circ}$ admits an exponential distribution with mean 1. Hence, given the wideband scenario, the decodable rate at any active user can be evaluated as

$$R[m] = B\mathbb{E}\left[\log_2\left(1+\frac{P}{|\mathcal{T}[m]|B}\eta_{u,k,\mathcal{U}[m]\setminus\mathcal{B}_\circ}\right)\right]. \quad (16)$$

As the length of each coded message is $F/\binom{U}{M}$, the duration of the mth time slot is given by

$$\tau[m] = \frac{F}{\binom{U}{M}}R[m]^{-1} = \frac{1}{\binom{U}{M}}\frac{F}{R[m]}. \quad (17)$$

If the schedule has T time slots, the overall delay of the proposed scheme is given by

$$D = \sum_{m=1}^{T}\tau[m] = \frac{1}{\binom{U}{M}}\sum_{m=1}^{T}\frac{F}{R[m]}. \quad (18)$$

The overall delay of the proposed scheme is generally larger than that of the SCK scheme due to four factors: (1) the requirement of having an integer number of time slots, (2) an excess number of time slots due to serving less than $M+L$ in some of the slots, (3) the possible power loss due to the transmission to incomplete groups, and finally (4) the rate reduction due to the loss of diversity. In the following, we analyze each of these overhead terms and show that the normalized number of excess time slots is negligible when the number of users in the network is large. At the same time, the effect of power loss and diversity will be negligible at a sufficiently large signal-to-noise ratio (SNR).

1) The Method Overhead: In an ideal scenario, where g groups are served in each time slot, the nominal number of time slots to serve all users in $\mathcal{G}$ is $T_\mathrm{N} = |\mathcal{G}|/g = \binom{U}{M+1}/g$. However, if T_N is not an integer, the minimal number of slots is $T_\mathrm{I} = \lceil T_\mathrm{N}\rceil = \lceil\binom{U}{M+1}/g\rceil$. We refer to

$$\zeta_\mathrm{M} = \frac{T_\mathrm{I}}{T_\mathrm{N}} - 1 = \frac{\lceil T_\mathrm{N}\rceil}{T_\mathrm{N}} - 1 \le \frac{1}{T_\mathrm{N}} \quad (19)$$

as *method overhead*. Note that $\zeta_\mathrm{M} = 0$ whenever g divides $\binom{U}{M+1}$. More importantly, as U grows, the method overhead converges to zero as $U^{-(M+1)}$.

2) The Algorithm Overhead: Recall that while the minimum length of the optimum scheduling is $T_\mathrm{I} = \lceil|\mathcal{G}|/g\rceil = \lceil\frac{M+1}{M+L}\binom{U}{M+1}\rceil$, an actual scheduling algorithm may require $T \ge T_\mathrm{I}$ time slots to serve all the groups. We define the algorithm overhead as $\zeta_\mathrm{A} = \frac{T-T_\mathrm{I}}{T_\mathrm{I}}$. It is shown in Section IV-A that for any integer g, there is a schedule with $T = T_\mathrm{I}$. Hence, when g is an integer, we have $\zeta_\mathrm{A} = 0$. On the other hand, the duration of a schedule obtained by the greedy algorithm is bounded by Theorem 2 (for some range of parameters). The following corollary bounds the algorithm overhead for the greedy scheduling algorithm, and its proof is presented in Appendix A.

Corollary 1: If g is an integer or $L = 2$, the greedy algorithm provides a schedule with an algorithm overhead $\zeta_\mathrm{A} = O(1/U)$. This overhead vanishes when U grows.

While we do not have an analytical guarantee for the greedy algorithm for non-integer values of g, our numerical results show a low overhead for all tested scenarios (see Fig. 2 in Section VI).

3) The Power Overhead: The proposed scheme with a scheduling table $\mathcal{T}$ serves $|\mathcal{T}[m]|$ groups in time slot m. Recall from (8) that the received SNR at an active user u in time slot m is given in (15), which is inversely proportional to the number of groups to be served.

For any ℓ, we define $R_\ell = B\mathbb{E}\left[\log_2(1+\frac{P}{\ell B}\eta)\right]$, where η admits an exponential distribution with mean 1. In an ideal scenario, (i.e., when g is an integer and $|\mathcal{T}[m]| = g$), we can achieve a rate of R_g. However, in general, we may serve up to $\lfloor g \rfloor + 2 > g$ groups (including $\lfloor g \rfloor$ groups to be completely served and a maximum of 2 partially served groups), and this leads to a power loss at the active users. We define the power overhead as $\zeta_{\mathrm{P}} \triangleq \frac{R_g}{\bar{R}} - 1$, where $\bar{R} = \left(\frac{1}{T}\sum_{m=1}^{T}\frac{1}{R[m]}\right)^{-1}$ is the harmonic mean of the rates achieved by the proposed scheme, and $R[m] = R_{|\mathcal{T}[m]|}$ is the rate achieved at time slot m. We also denote by $\gamma(\ell) := |\{m : |\mathcal{T}[m]| = \ell\}|$ the number of time slots during which exactly ℓ groups are served. In the following, we bound ζ_{P}. To this end, we need the following two lemmas, proved in Appendix B.

Lemma 1: For any $\ell \geq 0$, let $R_\ell = B\mathbb{E}\left[\log\left(1+\frac{P}{\ell B}\eta\right)\right]$, where η is an exponentially distributed random variable with $\mathbb{E}[\eta] = 1$. Then, R_ℓ is a decreasing functions of ℓ, and ℓR_ℓ is an increasing functions of ℓ.

Lemma 2: If the schedule obtained by Algorithm 1 has T rows, then $\gamma(\lfloor g \rfloor + 2) \leq \frac{\beta-1}{M+1}T$.

Now, we have

$$\begin{aligned}
\frac{R_g}{\bar{R}} &= \frac{1}{T}\sum_{m=1}^{T}\frac{R_g}{R_{|\mathcal{T}[m]|}} = \frac{1}{T}\sum_{\ell=1}^{\lfloor g \rfloor+2}\gamma(\ell)\frac{R_g}{R_\ell} \\
&= \frac{1}{T}\sum_{\ell=1}^{\lfloor g \rfloor+1}\gamma(\ell)\frac{R_g}{R_\ell} + \frac{\gamma(\lfloor g \rfloor+2)}{T}\frac{R_g}{R_{\lfloor g \rfloor+2}} \\
&\overset{(a)}{\leq} \frac{1}{T}\sum_{\ell=1}^{\lfloor g \rfloor+1}\gamma(\ell)\frac{R_g}{R_{\lfloor g \rfloor+1}} + \frac{\gamma(\lfloor g \rfloor+2)}{T}\frac{R_g}{R_{\lfloor g \rfloor+2}} \\
&\overset{(b)}{=} \frac{T-\gamma(\lfloor g \rfloor+2)}{T}\frac{R_g}{R_{\lfloor g \rfloor+1}} + \frac{\gamma(\lfloor g \rfloor+2)}{T}\frac{R_g}{R_{\lfloor g \rfloor+2}} \\
&\overset{(c)}{\leq} \frac{T-\gamma(\lfloor g \rfloor+2)}{T}\frac{\lfloor g \rfloor+1}{g} + \frac{\gamma(\lfloor g \rfloor+2)}{T}\frac{\lfloor g \rfloor+2}{g} \\
&= \frac{\lfloor g \rfloor+1}{g} + \frac{\gamma(\lfloor g \rfloor+2)/T}{g} \\
&\overset{(d)}{\leq} \frac{\lfloor g \rfloor+1}{g} + \frac{(\beta-1)/(M+1)}{g} \\
&\overset{(e)}{=} 1 + \frac{M}{(M+1)g} = 1 + \frac{M}{M+L},
\end{aligned} \tag{20}$$

where in (a) we used the first part of Lemma 1 for $\ell \leq \lfloor g \rfloor + 1$, (b) follows from the fact that $\sum_{\ell=1}^{\lfloor g \rfloor+2}\gamma(\ell) = T$, (c) holds since ℓR_ℓ is an increasing function of ℓ, due to Lemma 1, in (d) we used the fact that $\gamma(\lfloor g \rfloor + 2) \leq \frac{\beta-1}{M+1}T$ from Lemma 2, and finally, (e) holds since $\lfloor g \rfloor(M+1) + \beta = M + L$.

On the other, since $|\mathcal{T}[m]| \leq \lfloor g \rfloor + 2$, we have

$$\begin{aligned}
\bar{R} \geq R_{\lfloor g \rfloor+2} &= B\mathbb{E}\left[\log_2\left(1+\frac{P}{(\lfloor g \rfloor+2)B}\eta\right)\right] \\
&\geq B\mathbb{E}\left[\log_2\left(1+\frac{P}{gB}\eta\right)\right] - B\log_2\frac{\lfloor g \rfloor+2}{g} \\
&= R_g - B\log_2\frac{\lfloor g \rfloor+2}{g}.
\end{aligned} \tag{21}$$

Therefore, combining (20) and (21), we arrive at

$$\zeta_{\mathrm{P}} \leq \frac{R_g}{\bar{R}} - 1 \leq \min\left\{\frac{M}{M+L}, \frac{B}{R_{\lfloor g \rfloor+2}}\log_2\left(\frac{\lfloor g \rfloor+2}{g}\right)\right\}.$$

While $\frac{M}{M+L}$ is a universal upper bound for ζ_{P}, the power overhead vanishes as SNR increases:

$$\lim_{P\to\infty}\zeta_{\mathrm{P}} = \lim_{P\to\infty}\frac{B\log_2\left(\frac{\lfloor g \rfloor+2}{g}\right)}{R_{\lfloor g \rfloor+2}} = 0. \tag{22}$$

4) The Diversity Overhead: Note that the expectation in (16) is with respect to one exponentially distributed random variable $\eta_{u,k,\mathcal{U}[m]\setminus\mathcal{B}_\circ}$, which reflects the random fading in different frequency bins and the effective channel coefficients after zero forcing.

In contrast, the rate of the SCK scheme in (13) depends on the average of multiple η variables. This is due to the fact that an active user u is simultaneously decoding $\binom{M+L-1}{M}$ different messages, each pre-coded by a different beamforming vector (i.e., the coded message intended for users in group $\mathcal{V}$ (with $\mathcal{V} \subseteq \mathcal{Q}$, $|\mathcal{V}| = M+1$, and $u \in \mathcal{V}$) is transmitted in the direction of $\mathbf{h}_k^{\perp}[\![\mathcal{Q}\setminus\mathcal{V}]\!]$). Consequently, the rate of the SCK scheme depends on the average of identically distributed (and perhaps dependent) variables $\eta_{u,k,\mathcal{Q}\setminus\mathcal{V}}$ over the choice of $\mathcal{V}$.

Due to the concavity of the $\log_2(\cdot)$ function, the capacity of a Gaussian channel with a similar SNR provides an upper bound for both rates. Indeed, using the Jansen inequality, we can write

$$\begin{aligned}
R^{\mathsf{SCK}} &\leq B\log_2\left(1+\frac{P}{gB}\frac{1}{\binom{M+L-1}{M}}\sum_{\substack{\mathcal{V}:u\in\mathcal{V}\subseteq\mathcal{Q}\\|\mathcal{V}|=M+1}}\mathbb{E}\left[\eta_{u,k,\mathcal{Q}\setminus\mathcal{V}}\right]\right) \\
&= B\log_2\left(1+\frac{P}{gB}\right) \triangleq R^{\mathsf{AWGN}}.
\end{aligned} \tag{23}$$

A similar bound also holds for R_g, the rate achieved when exactly g groups are served in a time slot, that is, $R_g \leq R^{\mathsf{AWGN}}$. However, the gap between R^{SCK} and R^{AWGN} is smaller than the gap between R_g and R^{AWGN}, especially when the number of messages in the MAC, i.e. $\binom{M+L-1}{M}$, is large. This holds even though the variables $\eta_{u,k,\mathcal{Q}\setminus\mathcal{V}}$ in the summation are statistically dependent.

The effect of this averaging in the SCK scheme is similar to *receive diversity*; hence, the SCK scheme offers a higher rate compared to the proposed scheme. We use R^{AWGN} as a reference to compare the two rates. The following lemma provides a bound on the gap between R_g and R^{AWGN}. We present the proof of this lemma in Appendix B-C.

Lemma 3: Let η be an exponentially distributed random variable with $\mathbb{E}[\eta] = 1$. Then, we have $\mathbb{E}[\log(1+s\eta)] \geq 0.8\log(1+s)$ for every $s \geq 0$. Moreover, $\lim_{s\to 0}[\log(1+s) - \mathbb{E}[\log(1+s\eta)]] = 0.83$.

Using Lemma 3 for $s = P/gB$, we can bound the diversity overhead of the proposed scheme as

$$\begin{aligned}
\zeta_{\mathrm{D}} &\triangleq \frac{R^{\mathsf{SCK}}}{R_g} - 1 \leq \frac{R^{\mathsf{AWGN}}}{R_g} - 1 = \frac{B\log(1+P/gB)}{B\mathbb{E}[\log(1+P\eta/gB)]} - 1 \\
&= \frac{1}{0.8} - 1 \leq 0.25,
\end{aligned} \tag{24}$$

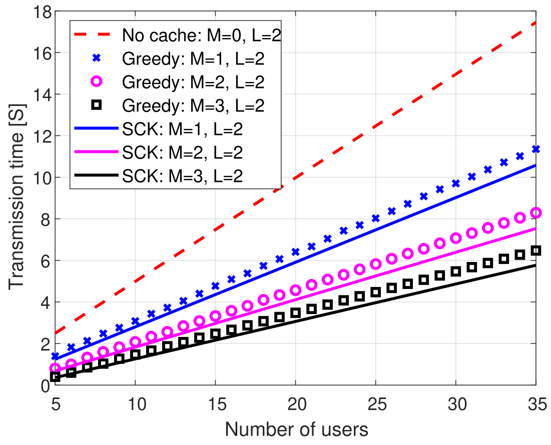

Fig. 1. Transmission time vs. the number of users. The performances of no cache, the proposed scheme with the greedy algorithm, and the SCK scheme [11] are compared. SNR is 35.7 dB, $F = 10$Mb and $B = 1$MHz.

for all values of $P \geq 0$. Moreover, for high SNR, we can use the second part of Lemma 3 to tighten this universal bound as

$$\lim_{P\to\infty} \zeta_{\mathrm{D}} = \lim_{P\to\infty} \frac{R^{\mathsf{SCK}}}{R_g} - 1 = \lim_{P\to\infty} \frac{R^{\mathsf{AWGN}}}{R_g} - 1 = \lim_{P\to\infty} \frac{0.83\ B}{B\mathbb{E}[\log_2(1+P\eta/gB)]} = 0. \tag{25}$$

This shows that the effect of channel diversity is negligible.

Next, having all the sources of loss analyzed, we can present a convenient comparison between the rate of the proposed scheme and that of the SCK scheme. Using (14) and (18) we can write

$$\frac{D}{D^{\mathsf{SCK}}} = \frac{\frac{1}{\binom{U}{M}}\sum_{m=1}^{T}\frac{F}{R[m]}}{\frac{U-M}{M+L}\frac{F}{R^{\mathsf{SCK}}}} = \frac{T}{\frac{U-M}{M+L}\binom{U}{M}} \frac{R_g}{\left(\frac{1}{T}\sum_{m=1}^{T}\frac{1}{R[m]}\right)^{-1}} \frac{R^{\mathsf{SCK}}}{R_g} = \frac{T_{\mathrm{I}}}{T_{\mathrm{N}}}\cdot\frac{T}{T_{\mathrm{I}}}\cdot\frac{R_g}{\bar{R}}\cdot\frac{R^{\mathsf{SCK}}}{R_g} = (1+\zeta_{\mathrm{M}})(1+\zeta_{\mathrm{A}})(1+\zeta_{\mathrm{P}})(1+\zeta_{\mathrm{D}}), \tag{26}$$

where we have $T_{\mathrm{N}} = \frac{M+1}{M+L}\binom{U}{M+1} = \frac{U-M}{M+L}\binom{U}{M}$ and $\bar{R} = \left(\frac{1}{T}\sum_{m=1}^{T}\frac{1}{R[m]}\right)^{-1}$. This bounds the ratio between the overall delay of the proposed scheme and that of the SCK scheme. As can be seen from (19) and Corollary 1, ζ_{M} and ζ_{A} are vanishing as U grows. Similarly, (22) and (25) imply that ζ_{P} and ζ_{D} become negligible at sufficiently large SNR.

VI. Numerical Results

This section shows that the proposed scheme can perform near optimally for the practical range of (M, L, U). We consider files of size $F = 10$ Mb, a communication bandwidth of $B = 1$ MHz, and a SNR of $P/B = 35.61$ dB. Thus, in the absence of cache (i.e., $M = 0$), using zero-forcing with $L = 2$ antennas at the transmitter we can achieve a per-user rate of $R = B\int_0^\infty \log_2(1 + 10^{3.561}\eta/2)e^{-\eta}d\eta \approx 10$ Mbps, and thus serve all users in $U/L = U/2$ seconds.

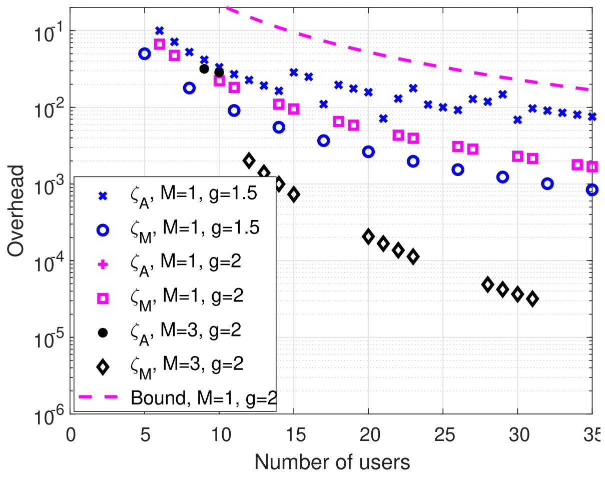

Fig. 2. Algorithm and method overheads as functions of the number of users. Due to the semi-logarithmic scale of the figure, any missing marker for $5 \leq U \leq 35$ should be interpreted as zero.

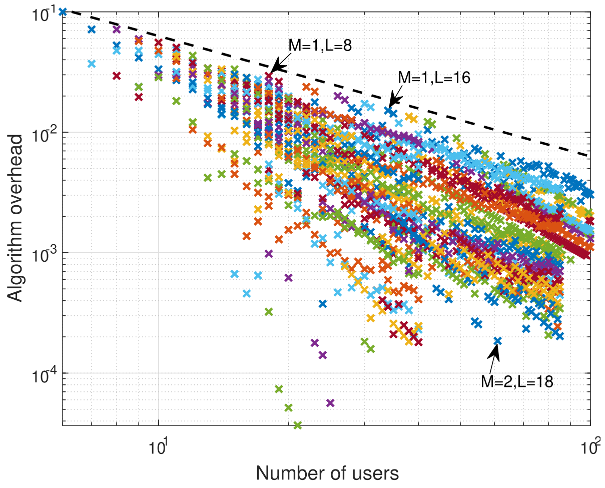

Fig. 3. Greedy algorithm overhead vs. the number of users. $1 \leq M \leq 10$ and $2 \leq L \leq 20$. Each color indicates a pair of (M, L). The figure shows all cases in which $U > 1.5(M + L)$ and $\binom{U}{M+1} \leq 10^5$. The dashed line depicts the $1/U$ slope (the line height is arbitrary).

Fig. 1 depicts the overall delay to serve all the users for different cache size values and the number of transmitter antennas. The SCK scheme is compared with the proposed scheme combined with the greedy scheduling algorithm presented in Section IV-B. It can be observed that cache-aided communication can significantly shorten the transmission delay, and except for a slight degradation, the proposed scheme performs similarly to the SCK scheme.

However, it is worth noting that the proposed scheme is more suitable for practical implementation compared to the SCK scheme (see Table II). For example, for $(U, M, L) = (30, 3, 4)$, the SCK scheme requires a division of each file into 10.5 million sub-packets, which is not feasible for a file of size 10 Mb. The subpacketization level for the proposed scheme under the same parameters is only 4060, which yields packets of size 2.46 Kb.

To illustrate the gap between the proposed and the SCK schemes, we depict the method overhead ζ_{M} and the algorithmic overhead ζ_{A} in terms of the number of users in Fig. 2. Note that these overhead terms do not depend on SNR. The figure presents ζ_{A} and ζ_{M} for the greedy algorithm,

TABLE II

A General Comparison of the Proposed Scheme in This Work and Other Schemes in the Literature. Here, T Denotes the Number of Time Slots for the Entire Course of Communication, and F Denotes the Subpacketization Level. The Number of Messages to Be Decoded by Each Active User in Each Time Slot and the Constraints on the System Parameters Are Listed in the Last Two Rows of the Table

	SCK [11]	Grouping [25]	Cyclic [26]	this work
T	$\binom{U}{M+L}$	$\binom{U/L}{M/L+1}$	$U(U-M)$	$\frac{M+1}{M+L}\binom{U}{M+1}$
F	$\binom{U}{M}\binom{U-M-1}{L-1}$	$\binom{U/L}{M/L}$	$(M+L)U$	$\binom{U}{M}$
ζ_{M}	$\zeta_{\mathrm{M}}=0$	$\zeta_{\mathrm{M}}=0$	$\zeta_{\mathrm{M}}=0$	$\zeta_{\mathrm{M}}\sim\frac{1}{U^{M+1}}$
ζ_{A}	$\zeta_{\mathrm{A}}=0$	$\zeta_{\mathrm{A}}=0$	$\zeta_{\mathrm{A}}=0$	$\zeta_{\mathrm{A}}=0$ (Optimum Alg.) $\zeta_{\mathrm{A}}\sim\frac{1}{U}$ (Greedy Alg.)
Low SNR ζ_{P}	$\zeta_{\mathrm{P}}=0$	$\zeta_{\mathrm{P}}\approx M$	$\zeta_{\mathrm{P}}\approx M$	$\zeta_{\mathrm{P}}\approx\frac{M}{M+L}$
ζ_{D}	$\zeta_{\mathrm{D}}=0$	$\zeta_{\mathrm{D}}\sim\frac{1}{\log P}$	$\zeta_{\mathrm{D}}\sim\frac{1}{\log P}$	$\zeta_{\mathrm{D}}\sim\frac{1}{\log P}$
# messages	$\binom{M+L-1}{M}$	1	1	1
Constraints	N.A.	$M/L\in\mathbb{Z}$	$L\geq M$	N.A.

as well as the analytical bound obtained in (28). It can be readily seen that both ζ_{M} and ζ_{A} decay rapidly with the number of users. The figure shows that the greedy algorithm is very efficient for both integer and non-integer values of g when the number of users is large.

Note that Fig. 2 is plotted in a semi-logarithmic scale. Hence, it cannot show the zero overheads, and any missing marker should be interpreted as 0. For instance, for $(U, M, L) = (15, 2, 1)$ we have $g = 1.5$ and $\binom{15}{2}/g$ is an integer, which leads to $\zeta_{\mathrm{M}} = 0$. Similarly, for $(M, L) = (1, 3)$ we have $g = 2$ and $\zeta_{\mathrm{A}} = 0$ for most of the values of U in the range $5 \leq U \leq 35$.

As we only have analytic bounds on the algorithm overhead for integer g, we also performed a massive simulation that tested many practical values of M and L. Fig. 3 depicts the algorithm overhead for the greedy algorithm for all values of $U \leq 100$, $1 \leq M \leq 10$ and $2 \leq L \leq 20$, for which the number of groups is bounded by[2] $\binom{U}{M+1} \leq 10^5$ and the number of users satisfy $U > 1.5(M + L)$. Recall that this is also shown on a logarithmic scale, and zeros are not shown.

The figure shows that the algorithm overhead is below 10% in all cases, and it decreases with the number of users. The dashed line depicts the $1/U$ slope (the line height is arbitrary). One can see that the $1/U$ scaling bounds all tested cases. Thus, the figure demonstrates the efficiency of the greedy algorithm for almost all scenarios of practical interest.

The power overhead ζ_{P} and the diversity overhead ζ_{D} versus SNR are shown in Fig. 4. Note that ζ_{P} and ζ_{D} do not depend on the number of users. While ζ_{P} is large for low SNR (especially when M is large and L is small), it significantly decays as the SNR grows and gets to less than 0.1 at 35 dB SNR. The diversity overhead, however, does not exceed 0.2 for the entire range of SNRs. It takes its maximum at ~ 10 dB, and decreases as SNR grows. Thus, both ζ_{P} and ζ_{D} become negligible at high SNR and do not affect the number of DoF.

Finally, in Fig. 5 we present the performance comparison (in terms of the overall communication delay) between the greedy algorithm (this work), the cyclic caching [26], and the SCK scheme [11], for two sets of parameters $(M, L, U) = (1, 3, 30)$ and $(M, L, U) = (2, 7, 15)$. In both settings, the number of groups to be served is an integer, namely, $g = 2$ and $g = 3$, respectively. We used $U = 30$ and $U = 15$ to avoid interfering curves for illustration purposes.

[2]The former condition limits the scheduling complexity to practical values. The latter condition is required for the algorithm to have sufficient diversity in the scheduling.

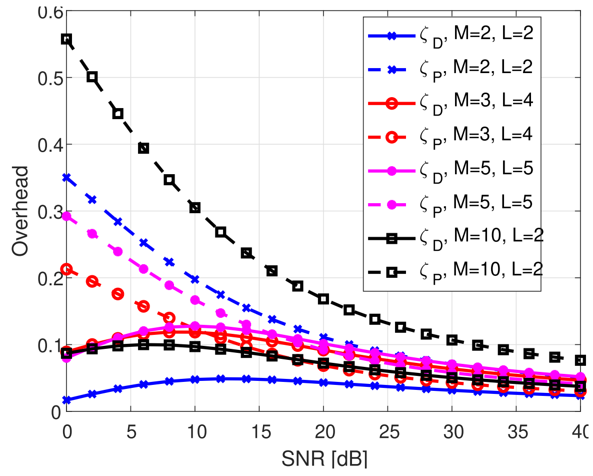

Fig. 4. Power overhead and diversity overhead (compared to those of [11] as the baseline) as a function of SNR for various scenarios.

The cyclic scheme has the lowest overall subpacketization but inferior performance in terms of the total transmission time. The SCK scheme completes serving all the users faster than the other two schemes. The small gaps between the curves for the proposed scheme and those of the SCK scheme account for the various overheads as characterized in (26). Note that the evaluation of (28) (see the proof of Corollary 1) leads to $\zeta_{\mathrm{A}} \leq 0.1379$ for $(M, L, U) = (1, 3, 30)$, and $\zeta_{\mathrm{A}} \leq 2.4066$ and $(M, L, U) = (2, 7, 15)$. However, even though the bound is not promising for $U = 15$, the actual overhead obtained by the greedy algorithm is much lower and leads to a small gap compared to the overall delay of the SCK scheme.

It can be seen that the overall delay of the cyclic scheme is larger than that of both other schemes. The primary reason for this gap is the fact that both the SCK and proposed schemes serve the users by broadcasting messages intended for $M + 1$ users, while the cyclic scheme transmits one message

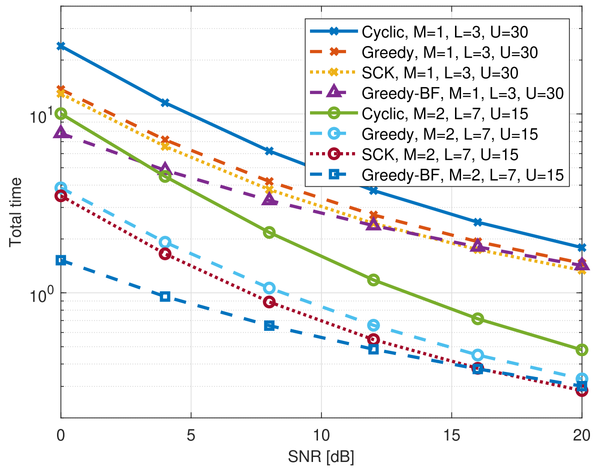

Fig. 5. Total communication delay vs. SNR, for the greedy algorithm (this work), the cyclic caching [26], and the SCK scheme [11].

per active user. This leads to a power loss of $M+1$ (i.e., $10\log_{10}(2) \approx 3$ and $10\log_{10}(3) \approx 4.78$ dB, for the two scenarios, respectively).

As promised earlier, Fig. 5 also demonstrates the additional gain that can be achieved by optimizing the number of used DoF and beamforming vectors as in [28]. This optimization can be applied to each of the three schemes but is shown only for the novel greedy algorithm (marked as *Greedy-BF* in the figure). As expected, such optimization has a significant gain at low SNR, but its gain decreases when the SNR increases. For example, for $M=2$ and $L=7$, at 0dB, the optimization gain is nearly 5dB, while at 20dB, the gain is only 1.5dB.

As the optimization gain is applicable to all considered schemes, it does not change our previous conclusions on the advantages of each scheme. Yet, the Greedy-BF curve emphasizes that at low SNR, it is important to also apply the optimization with respect to the number of DoF used and the beamforming vectors. Note that [27] takes this approach one step further, and the greedy algorithm is used to adapt the DoF at each time slot separately.

VII. Conclusion

We presented a novel scheme for cache-aided communication in cellular networks. The proposed scheme shows a significant increase in the network throughput compared to the no-cache case. The slight gap between the performance of the scheme and that of the SCK scheme vanishes as the number of users and SNR grow. However, the proposed scheme has low complexity and significantly reduces the subpacketization level. Unlike most previously known schemes, the presented scheme offers a practical solution that supports any integer M with any number of antennas and any number of users.

Appendix A
Analysis of the Greedy Algorithm

In this appendix, we prove Theorem 2 and Corollary 1. The core of the proofs is the fact that the greedy algorithm tries to serve g groups at a time as long as it can. This is possible for the majority of time slots and only towards the bottom of the table; some time slots might be wasteful, i.e., less than g groups are served during them. We first define a scheduling block and present two lemmas that express the formal statements of this claim. The proofs of the lemmas are given after the proofs of Theorem 2 and Corollary 1.

Definition 3: In an (M, L, U) system, a scheduling block *is a minimal-size collection of groups that can be* fully *served in a number of time slots.*

Note that for integer g, a scheduling block spans only one time slot, including g groups, while for $L=2$ a scheduling block spans $M+1$ time slots during which $M+2$ groups are served.

Lemma 4: Let $g = \frac{M+L}{M+1} \in \mathbb{N}$ be an integer, and $\tilde{\mathcal{G}} \subseteq \mathcal{G}$ be a collection of $(M+1)$-groups with

$$|\tilde{\mathcal{G}}| > f(U, M, g) \triangleq \binom{U}{M+1} - \binom{U-(g-1)(M+1)}{M+1}.$$

Then, for any score function, the greedy algorithm finds g pairwise disjoint groups. Moreover, we have

$$f(U, M, g) \leq \frac{(L-1)}{M!}U^M = c(M, g)U^M.$$

Lemma 5: Let $L=2$ and $\tilde{\mathcal{G}} \subseteq \mathcal{G}$ be any collection of $(M+1)$-groups with

$$|\tilde{\mathcal{G}}| > f(U, M, g) \triangleq \sum_{k=1}^{M} \frac{1}{k!}U^M.$$

Then, a score function exists for which the greedy algorithm finds a full scheduling block in $\tilde{\mathcal{G}}$. Moreover,

$$f(U, M, g) \leq (e-1)U^M = c(M, g).$$

Now, we are ready to present the proof of the theorem and its corollary.

Proof of Theorem 2: The greedy algorithm gradually selects groups to be served in each scheduling block according to some scoring function for the groups. Let s be the number of time slots in a scheduling block (where $s=1$ when g is an integer and $s=M+1$ for $L=2$). Also, let $\tilde{\mathcal{G}}[m]$ denote the set of groups that are not served by the end of the mth time slot. Then, $\tilde{\mathcal{G}}[0] = \mathcal{G}$ and the scheduling will be complete for some T, where $\tilde{\mathcal{G}}[T] = \emptyset$. Using Lemma 4 and Lemma 5, the greedy algorithm is guaranteed to find a scheduling block as long as $|\tilde{\mathcal{G}}[m]| > f(U, M, g)$. Note that as long as the algorithm can identify a scheduling block, no communication resource is wasted, and exactly $L+M$ users are served in each time slot of the block, leading to a total of $s(M+L)$ users, and equivalently sg groups to be completely served during a block. Thus, after the bth scheduling block we have $|\tilde{\mathcal{G}}[bs]| = |\mathcal{G}| - bsg$, provided that $|\tilde{\mathcal{G}}[bs]| > f(U, M, g)$. Assume b_0 scheduling blocks are generated, and no more blocks can be completed by the algorithm. Then,

$$\begin{aligned} |\mathcal{G}| - (b_0-1)sg = |\tilde{\mathcal{G}}[(b_0-1)s]| &> f(U, M, G) \\ &\geq |\tilde{\mathcal{G}}[b_0 s]| = |\mathcal{G}| - b_0 sg, \end{aligned} \quad (27)$$

which implies $b_0 = \left\lceil \frac{\binom{U}{M+1} - f(U,M,g)}{sg} \right\rceil$.

For a time slot $m > sb_0$, the algorithm may need to schedule the groups less efficiently. In the worst-case scenario, we may serve the remaining groups one per time slot, requiring an additional $|\tilde{\mathcal{G}}[b_0 s]|$ number of time slots. Hence, the number of time slots in the schedule is upper-bounded by

$$T \le b_0 s + |\tilde{\mathcal{G}}[b_0 s]| \le \left\lceil \frac{\binom{U}{M+1} - f(U,M,g)}{sg} \right\rceil s + f(U,M,g)$$
$$\le \frac{1}{g}\binom{U}{M+1} + \frac{g-1}{g} f(U,M,g) + s.$$

Replacing the value of s for the two scenarios, we conclude the proof of Theorem 2. ■

Proof of Corollary 1: First note that since

$$T_{\mathrm{I}} = \lceil T_{\mathrm{N}} \rceil \ge T_{\mathrm{N}} = \frac{1}{g}\binom{U}{M+1},$$

we can write $\zeta_{\mathrm{A}} = \frac{T - T_{\mathrm{I}}}{T_{\mathrm{I}}} \le \frac{T - T_{\mathrm{N}}}{T_{\mathrm{N}}}$. Then, using the bound in Theorem 2, we have

$$\zeta_{\mathrm{A}} \le \frac{T - T_{\mathrm{N}}}{T_{\mathrm{N}}} \le \frac{\frac{g-1}{g} f(U,M,g) + s}{\frac{1}{g}\binom{U}{M+1}}$$
$$\le \frac{(g-1)c(M,g)U^M + sg}{\left(\frac{U}{M+1}\right)^{M+1}} = O\left(\frac{1}{U}\right). \quad (28)$$

Here, $c(M,g) = \frac{L-1}{M!}$ for integer g and $c(M,g) = e-1$ for $L=2$. This completes the proof.

Proof of Lemma 4: We prove the lemma by induction on g. First, for $g=1$, we have $f(U,M,g)=0$, and the claim holds true. Next, assume that the claim holds for $g-1$. Then, since $|\tilde{\mathcal{G}}| > f(U,M,g) > f(U,M,g-1)$, the induction assumption implies that the greedy algorithm will succeed in choosing $g-1$ non-overlapping groups. We denote the set of these non-overlapping groups by $\mathcal{Q}_{g-1}$. We show that if the condition of the lemma holds, we can always find a gth disjoint group to be added to $\mathcal{Q}_{g-1}$ to form $\mathcal{Q}_g$. The selected $g-1$ groups include a total of $(g-1)(M+1) = L-1$ users. The gth group should be disjoint from all groups in $\mathcal{Q}_{g-1}$; hence, its elements should be chosen from the remaining $U-(L-1)$ users. Let us denote by $\mathcal{A}_g \subseteq \mathcal{G}$ the set of all groups which are disjoint from all groups in $\mathcal{Q}_{g-1}$, where $|\mathcal{A}_g| = \binom{U-(g-1)(M+1)}{M+1}$. Since $\mathcal{A}_g, \tilde{\mathcal{G}} \subseteq \mathcal{G}$, we have $|\mathcal{A}_g \cup \tilde{\mathcal{G}}| \le |\mathcal{G}| = \binom{U}{M+1}$. Therefore,

$$|\mathcal{A}_g \cap \tilde{\mathcal{G}}| = |\mathcal{A}_g| + |\tilde{\mathcal{G}}| - |\mathcal{A}_g \cup \tilde{\mathcal{G}}|$$
$$> \binom{U-(g-1)(M+1)}{M+1} + f(U,M,g) - \binom{U}{M+1} = 0.$$

This implies that there is at least one group from $\mathcal{A}_g$ that appears in $\tilde{\mathcal{G}}$ and can be identified by the greedy algorithm. Appending that groups to $\mathcal{Q}_{g-1}$, we obtain $\mathcal{Q}_g$. The inequality in the lemma stems from Pascal's identity:

$$\binom{U}{M+1} - \binom{U-(L-1)}{M+1} = \sum_{t=1}^{L-1}\binom{U-t}{M}$$
$$\le (L-1)\binom{U-1}{M} \le \frac{(L-1)U^M}{M!}.$$

■

Proof of Lemma 5: Note that when $L=2$, we can serve (up to) $M+2$ users at a time, which includes a completely-served group, as well as one user from a partially-served group. In order to fully serve the partially-served group (of size $M+1$), we need $M+1$ time slots. Let $\mathcal{A}_1, \mathcal{A}_2, \dots, \mathcal{A}_{M+1}$ be the completely-served groups in in the block and $\mathcal{A}_0$ be the partially-served group. From the definition of *scheduling* (see Definition 3), we have $|\mathcal{A}_0 \cap \mathcal{A}_m| = M$ for $m = 1, 2\dots, M+1$. Let us define the scoring function

$$\mathsf{Scr}(\mathcal{A}, \tilde{\mathcal{G}}) \triangleq \min_{a \in \mathcal{A}} \left| \left\{ \mathcal{B} \in \tilde{\mathcal{G}} : \mathcal{A} \setminus \mathcal{B} = \{a\} \right\} \right|. \quad (29)$$

If there exists any group $\mathcal{A}_0 \in \tilde{\mathcal{G}}$ with $\mathsf{Scr}(\mathcal{A}, \tilde{\mathcal{G}}) > 0$, then $\tilde{\mathcal{G}}$ contains a valid scheduling block with $\mathcal{A}_0$ as its partially-served group. It remains to show the existence of such an $\mathcal{A}_0$.

We prove the lemma by induction over M. First, let $M=1$ (where $\sum_{k=1}^{1} 1/k! = 1$) and consider a collection of $|\tilde{\mathcal{G}}| \ge U$ groups. Consider the graph over $([U], \tilde{\mathcal{G}})$, that is, the graph whose nodes are the elements of $[U]$ and its edges are the groups in $\tilde{\mathcal{G}}$. Since $|\tilde{\mathcal{G}}| \ge U = |[U]|$ (i.e., the number of edges is not smaller than the number of nodes), the graph has at least one cycle. If the cycle is of length 3, say $\{\{a,b\},\{b,c\},\{c,a\}\} \subseteq \tilde{\mathcal{G}}$, then the scheduling block consists of $\mathcal{T}[m] = (\{c\},\{a,b\})$ and $\mathcal{T}[m+1] = (\{b\},\{c,a\})$. Otherwise, we have a cycle of length more than 3, which includes a path $\{\{a,b\},\{b,c\},\{c,d\}\} \subseteq \tilde{\mathcal{G}}$. In this case, the scheduling block is formed as $\mathcal{T}[m] = (\{c\},\{a,b\})$ and $\mathcal{T}[m+1] = (\{b\},\{c,d\})$.

Next, we assume that the claim holds for $M = m-1$, and prove it for $M=m$. Consider an $(M=m, L=2, U)$ systems, and let $\tilde{\mathcal{G}}$ be a collection of $(m+1)$-groups with $|\tilde{\mathcal{G}}| \ge U^m \sum_{k=1}^{m} 1/k!$. We partition $\tilde{\mathcal{G}}$ into non-overlapping sets $\tilde{\mathcal{G}}_u = \{\mathcal{A} \in \tilde{\mathcal{G}} : \min(\mathcal{A}) = u\}$, for $u \in [U]$. Also define $\tilde{\mathcal{G}}'_u = \{\mathcal{A} \setminus \{u\} : \mathcal{A} \in \tilde{\mathcal{G}}_u\}$ for $u \in [U]$. Note that the elements of $\tilde{\mathcal{G}}'_u$ are groups of size m, and we have $\sum_{u=1}^{U} |\tilde{\mathcal{G}}'_u| = \sum_{u=1}^{U} |\tilde{\mathcal{G}}_u| = |\tilde{\mathcal{G}}| \ge c_m U^m$. However, there might be an overlap between $\tilde{\mathcal{G}}'_u$ and $\tilde{\mathcal{G}}'_v$. Define $\mathcal{S}_{u,v} = \tilde{\mathcal{G}}'_u \cap \tilde{\mathcal{G}}'_v$, and let $u^\star = \arg\max_u \left|\bigcup_{v:v\neq u} \mathcal{S}_{u,v}\right|$ and $\tilde{\mathcal{G}}' = \bigcup_{v:v\neq u^\star} \mathcal{S}_{u^\star,v}$. We can write

$$\sum_{k=1}^{m} \frac{1}{k!} U^m \le |\tilde{\mathcal{G}}| = \left| \bigcup_{u=1}^{U} \tilde{\mathcal{G}}'_u \right|$$
$$= \left| \left(\bigcup_{u=1}^{U} \left(\tilde{\mathcal{G}}'_u \setminus \left(\bigcup_{v:v\neq u} \mathcal{S}_{u,v} \right) \right) \right) \cup \left(\bigcup_{u=1}^{U} \left(\bigcup_{v:v\neq u} \mathcal{S}_{u,v} \right) \right) \right|$$
$$\le \left| \bigcup_{u=1}^{U} \left(\tilde{\mathcal{G}}'_u \setminus \left(\bigcup_{v:v\neq u} \mathcal{S}_{u,v} \right) \right) \right| + \sum_{u=1}^{U} \left| \bigcup_{v:v\neq u} \mathcal{S}_{u,v} \right|$$
$$\overset{(a)}{\le} \binom{U}{m} + U \cdot |\tilde{\mathcal{G}}'| \le \frac{U^m}{m!} + U \cdot |\tilde{\mathcal{G}}'|,$$

where (a) holds since $\bigcup_{u=1}^{U} \left(\tilde{\mathcal{G}}'_u \setminus \left(\bigcup_{v:v\neq u} \mathcal{S}_{u,v} \right) \right)$ is a non-repeating collection of groups of size m, and $\left| \bigcup_{v:v\neq u} \mathcal{S}_{u,v} \right| \le \left| \bigcup_{v:v\neq u^\star} \mathcal{S}_{u^\star,v} \right| = |\tilde{\mathcal{G}}'|$ for every $u \in [U]$.

The inequality above leads to

$$|\tilde{\mathcal{G}}'| \ge \frac{1}{U} \left(\sum_{k=1}^{m} \frac{1}{k!} - \frac{1}{m!} \right) U^m = U^{m-1} \sum_{k=1}^{m-1} \frac{1}{k!}.$$

Hence, the induction assumption implies that a scheduling block exists from the groups in $\tilde{\mathcal{G}}'$. Let $\mathcal{A}'_0, \mathcal{A}'_1, \dots, \mathcal{A}'_m$ be the groups that form a scheduling block in $\tilde{\mathcal{G}}'$ (where $\mathcal{A}'_0$ is the partially-served group). Recall that $\tilde{\mathcal{G}}' = \bigcup_{v:v\neq u^\star} \mathcal{S}_{u^\star,v}$, and hence every group in $\tilde{\mathcal{G}}'$ is subset of $[U] \setminus \{u^\star\}$ of size m. Moreover, since $\mathcal{A}_0 \in \tilde{\mathcal{G}}'$, there exists some $v^\star$ such that $\mathcal{A}_0 \in \mathcal{S}_{u^\star,v^\star} = \tilde{\mathcal{G}}'_{u^\star} \cap \tilde{\mathcal{G}}'_{v^\star}$. Now consider groups $\mathcal{A}_i = \mathcal{A}'_i \cup \{u^\star\}$ for $i = 0, \dots, m$, and $\mathcal{A}_{m+1} = \mathcal{A}'_0 \cup \{v^\star\}$. It is easy to verify that $\mathcal{A}_0, \mathcal{A}_1, \dots, \mathcal{A}_m \in \tilde{\mathcal{G}}_{u^\star} \subseteq \tilde{\mathcal{G}}$ and $\mathcal{A}_{m+1} \in \tilde{\mathcal{G}}_{v^\star} \subseteq \tilde{\mathcal{G}}$. Moreover, for $i = 1, \dots, m$ we have

$$\begin{aligned}|\mathcal{A}_i \cap \mathcal{A}_0| &= |(\mathcal{A}'_i \cup \{u^\star\}) \cap (\mathcal{A}'_0 \cup \{u^\star\})| \\ &= |\mathcal{A}'_i \cap \mathcal{A}'_0| + |\{u^\star\}| = (m-1) + 1 = m.\end{aligned}$$

Finally, we have

$$|\mathcal{A}_{m+1} \cap \mathcal{A}_0| = |(\mathcal{A}'_0 \cup \{v^\star\}) \cap (\mathcal{A}'_0 \cup \{u^\star\})| = |\mathcal{A}'_0| = m.$$

These together imply that $\mathsf{Scr}(\mathcal{A}_0, \tilde{\mathcal{G}}) > 0$. This completes the proof of the lemma. ■

Appendix B
Proof of the Auxiliary Lemmas

A. Proof of Lemma 1

The proof of the first claim follows from the facts that $\frac{P}{\ell}$ is increasing with ℓ, and $\log(\cdot)$ is an increasing function. To show the second claim, we use the fact that $1 + ax \leq (1+x)^a$ for every $x \geq 0$ and $a \geq 1$, or equivalently, $\log(1+ax) \leq a \log(1+x)$. Letting $\ell_1 \leq \ell_2$, and plugging $x = P\eta/\ell_2 B$ and $a = \frac{\ell_2}{\ell_1} \geq 1$ we get $\log(1 + (\ell_2/\ell_1)(P\eta/\ell_2 B)) \leq (\ell_2/\ell_1) \log(1 + P\eta/\ell_2 B)$. Multiplying both sides by $B\ell_1$, taking expectation with respect to η, we get $\ell_1 R_{\ell_1} \leq \ell_2 R_{\ell_2}$. ■

B. Proof of Lemma 2

First note that if $\beta = 0$, then g is an integer, and we always serve (at most) g groups at any given time, and hence, $\gamma(\lfloor g \rfloor + 2) = 0$. Next, we focus on the regimes in which we have $\beta \geq 1$.

Consider the first $M+1$ rows of the scheduling table $\mathcal{T}$. Recall that each row can fit up to β users from the partially served groups. Hence, the total number of users served as a part of the partially served groups in the first $M+1$ time slots is most $\beta(M+1)$, and thus, the number of partially served groups contributing in these $M+1$ time slots is most β. Without loss of generality, let us denote these groups by $\mathcal{B}_1, \mathcal{B}_2, \dots, \mathcal{B}_\beta$. For a time slot m, we have $|\mathcal{T}[m]| = \lfloor g \rfloor + 2$, if and only if the algorithm starts row m with $|\mathsf{Buffer}| < \beta$ users from a partially served group $\mathcal{B}_i$, continues with a selection of $\lfloor g \rfloor$ complete groups, it finishes the mth row by another partially serving group $\mathcal{B}_{i+1}$ for some $i = 1, 2, \dots, \beta - 1$. Hence, the number of such m's is at most $\beta - 1$, which implies at most $(\beta-1)/(M+1)$ fraction of the first $M+1$ rows can serve $\lfloor g \rfloor + 2$ groups. The same argument can be applied to any other $M+1$ consecutive rows of the table and implies $\gamma(\lfloor g \rfloor + 2) \leq \frac{\beta-1}{M+1} T$. ■

C. Proof of Lemma 3

Define $f(s) := e^{-1/s} \ln 2 \left(\mathbb{E}\left[\log(1+s\eta)\right] - 0.8 \log(1+s)\right)$, where η admits an exponential distribution with $\mathbb{E}[\eta] = 1$. We aim to show that $f(s) \geq 0$ for every $s \geq 0$. We start with

$$\begin{aligned}\mathbb{E}[\log(1+s\eta)] &= \int_0^\infty \log(1+s\eta) e^{-\eta} d\eta \\ &\overset{(a)}{=} -\log(1+s\eta)e^{-\eta}\Big|_0^\infty - \int_0^\infty \frac{-se^{-\eta}}{(1+s\eta)\ln 2} d\eta \\ &\overset{(b)}{=} \frac{1}{\ln 2} \int_{1/s}^\infty \frac{e^{-(t-1/s)}}{t} dt = \frac{e^{1/s}}{\ln 2} \int_{1/s}^\infty \frac{e^{-t}}{t} dt,\end{aligned}$$

where in (a) we used integration by parts with $u = \log(1+s\eta)$, $dv = e^{-\eta} d\eta$ and $v = -e^{-\eta}$, and (b) is due to a change of variable $t = \eta + \frac{1}{s}$. Hence, we have

$$f(s) = \int_{1/s}^\infty \frac{e^{-t}}{t} dt - 0.8\ e^{-1/s} \ln(1+s).$$

Taking derivative of $f(s)$, we get

$$\frac{df(s)}{ds} = \frac{e^{-1/s}}{s^2} \left(s - 0.8 \ln(1+s) - 0.8 \frac{s^2}{1+s}\right). \tag{30}$$

Now, for function $g(s) := s - 0.8 \ln(1+s) - 0.8 \frac{s^2}{1+s}$ we have

$$rCl \frac{dg(s)}{ds} = 1 - 0.8 \frac{1}{1+s} - 0.8 \frac{s^2 + 2s}{(1+s)^2} = \frac{(s-1)^2}{5(s+1)^2} \geq 0.$$

This implies that $g(s)$ is a non-decreasing function, and thus $g(s) \geq g(0) = 0$ for every $s \geq 0$. This, together with (30), lead to $\frac{df(s)}{ds} \geq 0$, i.e., $f(s)$ is also a non-decreasing function. Note that $\lim_{s\to 0} f(s) = 0$, and thus, $f(s) \geq 0$ for every $s \geq 0$.

On the other hand, we have

$$\begin{aligned}&\lim_{s\to\infty} [\log(1+s) - \mathbb{E}[\log(1+s\eta)]] \\ &= \lim_{s\to\infty} \left[\log(1+s) - \int_0^\infty \log(1+s\eta) e^{-\eta} d\eta\right] \\ &= \lim_{s\to\infty} \int_0^\infty \log\left(\frac{1+s}{1+s\eta}\right) e^{-\eta} d\eta \\ &\overset{(c)}{=} \int_0^\infty \lim_{s\to\infty} \log\left(\frac{1+s}{1+s\eta}\right) e^{-\eta} d\eta = -\int_0^\infty \log \eta \cdot e^{-\eta} d\eta = 0.833,\end{aligned}$$

where (c) follows from the dominated convergence theorem (with $|\log(1+s)/(1+s\eta) \cdot e^{-\eta}|$ being dominated by the integrable function $\log 1/\eta \cdot e^{-\eta}$). This completes the proof of the lemma. ■

References

[1] S. Mohajer and I. Bergel, "MISO cache-aided communication with reduced subpacketization," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Jun. 2020, pp. 1–6, doi: 10.1109/ICC40277.2020.9149433.

[2] I. Bergel and S. Mohajer, "Practical scheme for MISO cache-aided communication," in *Proc. IEEE 21st Int. Workshop Signal Process. Adv. Wireless Commun. (SPAWC)*, May 2020, pp. 1–5, doi: 10.1109/SPAWC48557.2020.9154258.

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

[4] S. S. Bidokhti, M. Wigger, and A. Yener, "Gaussian broadcast channels with receiver cache assignment," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2017, pp. 1–6.

[5] M. Gregori, J. Gómez-Vilardebò, J. Matamoros, and D. Gündüz, "Joint transmission and caching policy design for energy minimization in the wireless backhaul link," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jun. 2015, pp. 1004–1008.

[6] S. Yang, K.-H. Ngo, and M. Kobayashi, "Content delivery with coded caching and massive MIMO in 5G," in *Proc. 9th Int. Symp. Turbo Codes Iterative Inf. Process. (ISTC)*, Sep. 2016, pp. 370–374.

[7] S. S. Bidokhti, M. Wigger, and R. Timo, "Noisy broadcast networks with receiver caching," *IEEE Trans. Inf. Theory*, vol. 64, no. 11, pp. 6996–7016, Nov. 2018.

[8] S. S. Bidokhti, M. Wigger, and R. Timo, "Erasure broadcast networks with receiver caching," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jul. 2016, pp. 1819–1823.

[9] A. Ghorbel, M. Kobayashi, and S. Yang, "Content delivery in erasure broadcast channels with cache and feedback," *IEEE Trans. Inf. Theory*, vol. 62, no. 11, pp. 6407–6422, Nov. 2016.

[10] S. P. Shariatpanahi, S. A. Motahari, and B. H. Khalaj, "Multi-server coded caching," *IEEE Trans. Inf. Theory*, vol. 62, no. 12, pp. 7253–7271, Dec. 2016.

[11] S. P. Shariatpanahi, G. Caire, and B. H. Khalaj, "Multi-antenna coded caching," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jun. 2017, pp. 2113–2117.

[12] K.-H. Ngo, S. Yang, and M. Kobayashi, "Scalable content delivery with coded caching in multi-antenna fading channels," *IEEE Trans. Wireless Commun.*, vol. 17, no. 1, pp. 548–562, Jan. 2018.

[13] S. Mohajer, I. Bergel, and G. Caire, "Cooperative wireless mobile caching: A signal processing perspective," *IEEE Signal Process. Mag.*, vol. 37, no. 2, pp. 18–38, Mar. 2020.

[14] A. M. Ibrahim, A. A. Zewail, and A. Yener, "Optimization of heterogeneous caching systems with rate limited links," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2017, pp. 1–6.

[15] E. Lampiris, A. Bazco-Nogueras, and P. Elia, "Resolving the feedback bottleneck of multi-antenna coded caching," *IEEE Trans. Inf. Theory*, vol. 68, no. 4, pp. 2331–2348, Apr. 2022.

[16] L. Tang and A. Ramamoorthy, "Coded caching schemes with reduced subpacketization from linear block codes," *IEEE Trans. Inf. Theory*, vol. 64, no. 4, pp. 3099–3120, Apr. 2018.

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

[18] C. Shangguan, Y. Zhang, and G. Ge, "Centralized coded caching schemes: A hypergraph theoretical approach," *IEEE Trans. Inf. Theory*, vol. 64, no. 8, pp. 5755–5766, Aug. 2018.

[19] S. Jin, Y. Cui, H. Liu, and G. Caire, "A new order-optimal decentralized coded caching scheme with good performance in the finite file size regime," *IEEE Trans. Commun.*, vol. 67, no. 8, pp. 5297–5310, Aug. 2019.

[20] S. Jin, Y. Cui, H. Liu, and G. Caire, "Uncoded placement optimization for coded delivery," in *Proc. 16th Int. Symp. Model. Optim. Mobile, Ad Hoc, Wireless Netw. (WiOpt)*, May 2018, pp. 1–8.

[21] M. Cheng, Q. Yan, X. Tang, and J. Jiang, "Coded caching schemes with low rate and subpacketizations," 2017, *arXiv:1703.01548*.

[22] K. Shanmugam, A. M. Tulino, and A. G. Dimakis, "Coded caching with linear subpacketization is possible using ruzsa-Szeméredi graphs," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jun. 2017, pp. 1237–1241.

[23] L. Tang and A. Ramamoorthy, "Low subpacketization schemes for coded caching," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jun. 2017, pp. 2790–2794.

[24] X. Zhang, N. Woolsey, and M. Ji, "Cache-aided interference management using hypercube combinatorial design with reduced subpacketizations and order optimal sum-degrees of freedom," *IEEE Trans. Wireless Commun.*, vol. 20, no. 8, pp. 4797–4810, Aug. 2021.

[25] E. Lampiris and P. Elia, "Adding transmitters dramatically boosts coded-caching gains for finite file sizes," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 6, pp. 1176–1188, Jun. 2018.

[26] M. Salehi, E. Parrinello, S. P. Shariatpanahi, P. Elia, and A. Tölli, "Low-complexity high-performance cyclic caching for large MISO systems," *IEEE Trans. Wireless Commun.*, vol. 21, no. 5, pp. 3263–3278, May 2022.

[27] I. Bergel and S. Mohajer, "Channel aware greedy algorithm for MISO cache-aided communication," in *Proc. IEEE 23rd Int. Workshop Signal Process. Adv. Wireless Commun. (SPAWC)*, Jul. 2022, pp. 1–5.

[28] A. Tölli, S. P. Shariatpanahi, J. Kaleva, and B. H. Khalaj, "Multi-antenna interference management for coded caching," *IEEE Trans. Wireless Commun.*, vol. 19, no. 3, pp. 2091–2106, Mar. 2020.

[29] Z. Baranyai, "On the factorization of the complete uniform hypergraph," in *Colloquia Mathematica Societatis János Bolyai. Infinite and Finite Sets, Keszthely (Hungary), 1973*, vol. 10, A. Hajnal, R. Rado, and V. T. Sós, Eds. Amsterdam, The Netherlands: North Holland, pp. 91–107.

Soheil Mohajer (Member, IEEE) received the B.Sc. degree in electrical engineering from the Sharif University of Technology, Tehran, Iran, in 2004, and the M.Sc. and Ph.D. degrees in communication systems from the École Polytechnique Fédérale de Lausanne (EPFL), Switzerland, in 2005 and 2010, respectively. He was a Post-Doctoral Researcher with Princeton University, Princeton, NJ, USA, from 2010 to 2011, and the University of California at Berkeley from 2011 to 2013. He is currently a McKnight Land-Grant Associate Professor with the Department of Electrical and Computer Engineering, University of Minnesota, Twin Cities. His research interests include information theory and its applications in distributed storage systems, data delivery networks, distributed optimization, and statistical machine learning. He received the NSF CAREER Award in 2018. He has served as an Editor for IEEE TRANSACTIONS ON COMMUNICATIONS from 2018 to 2021.

Itsik Bergel (Senior Member, IEEE) received the B.Sc. degree in electrical engineering and the B.Sc. degree in physics from the Ben-Gurion University of the Negev, Israel, in 1993 and 1994, respectively, and the M.Sc. and Ph.D. degrees in electrical engineering from the University of Tel Aviv, Tel-Aviv, Israel, in 2000 and 2005, respectively. From 2001 to 2003, he was a Senior Researcher with the Intel Communications Research Laboratory. In 2005, he was a Post-Doctoral Researcher with the Dipartimento di Elettronica, Politecnico di Torino, Turin, Italy. He is currently an Associate Professor with the Faculty of Engineering, Bar-Ilan University, Ramat Gan, Israel. His main research interests include interference mitigation in wireless communications, cooperative transmission in cellular networks, and coded caching. He was an Associate Editor for the IEEE TRANSACTIONS ON SIGNAL PROCESSING. He is currently an Editor of the IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS.