Low-Complexity High-Performance Cyclic Caching for Large MISO Systems

MohammadJavad Salehi, *Member, IEEE*, Emanuele Parrinello, *Member, IEEE*, Seyed Pooya Shariatpanahi, Petros Elia, and Antti Tölli, *Senior Member, IEEE*

***Abstract*— Multi-antenna coded caching is known to combine a global caching gain that is proportional to the cumulative cache size found across the network, with an additional spatial multiplexing gain that stems from using multiple transmitting antennas. However, a closer look reveals two severe bottlenecks; the well-known exponential subpacketization bottleneck that dramatically reduces performance when the communicated file sizes are finite, and the considerable optimization complexity of beamforming multicast messages when the SNR is finite. We here present an entirely novel caching scheme, termed *cyclic multi-antenna coded caching*, whose unique structure allows for the resolution of the above bottlenecks in the crucial regime of many transmit antennas. For this regime, where the multiplexing gain can exceed the coding gain, our new algorithm is the first to achieve the exact one-shot linear optimal DoF with a subpacketization complexity that scales only linearly with the number of users, and the first to benefit from a multicasting structure that allows for exploiting uplink-downlink duality in order to yield optimized beamformers ultra-fast. In the end, our novel solution provides excellent performance for networks with finite SNR, finite file sizes, and many users.**

***Index Terms*— Coded caching, multi-antenna communication, low-subpacketization, optimized beamforming, finite-SNR.**

I. Introduction

WIRELESS communication networks are under mounting pressure to support the exponentially increasing volumes of multimedia content, as well as to support the imminent emergence of applications such as wireless immersive viewing and reliable autonomous driving [1]. For the efficient delivery of such multimedia content, the work of Maddah-Ali and Niesen [2] proposed the idea of *coded caching* as a means of increasing the data rates by exploiting cache content across the network. This approach considered a single-stream (single-antenna) downlink network of cache-enabled receiving users who can pre-fetch data into their cache memories, in a way that — during the subsequent content delivery — the achievable rates can be boosted by a multiplicative factor that is proportional to the cumulative cache size across the entire network. The key to achieving this speedup was the ability of coded caching to successfully multicast individual messages to many users at a time, such that each user can use its cache content to remove unwanted messages from the received signal. In this context, a system that can multicast to $t+1$ users at a time is said to enjoy a (cache-aided) Degrees-of-Freedom (DoF) performance $t+1$, which also matches the aforementioned multiplicative speedup factor in the high-SNR regime. This quantity t — which is often referred to as the (additive) *coded caching gain* — depends on the number of receiving users and the size of their cache, and it effectively describes the redundancy with which data can be cached across the network.

Motivated by the unavoidable dominance of multi-antenna paradigms in wireless communications [3], Shariatpanahi *et al.* [4], [5] explored the cache-aided multi-antenna setting, for which it revealed that this same caching gain could, in fact, be maintained in its fullest. In a basic downlink scenario with L such transmit antennas, the work in [5] developed a method that achieved a DoF of $t+L$, which was shown to be optimal under basic assumptions of uncoded cache placement and one-shot data delivery (cf. [6]).

However, despite the original theoretical promises for large caching gains, in reality, coded caching can suffer from severe bottlenecks that dramatically limit these gains. Undoubtedly the most damaging of these is the well-known *subpacketization bottleneck*, which stems from the fact that the aforementioned caching gains require (cf. [7], [8]) that each file is split into a number of subfiles that scales exponentially with the number of receiving users. This requirement is exacerbated in multi-antenna coded caching approaches (cf. [5]), where the subpacketization (and thus the file-size requirements) can far exceed those of the original scheme in [2]. In essence, subpacketization requirements rendered coded caching hard to implement in most moderate- or large-sized networks.

While though the subpacketization requirements for achieving full caching gains in the single antenna setting are indeed

Manuscript received September 25, 2020; revised February 26, 2021 and July 9, 2021; accepted October 5, 2021. Date of publication October 20, 2021; date of current version May 10, 2022. This work was supported in part by the Academy of Finland under Grant 319059 (Coded Collaborative Caching for Wireless Energy Efficiency) and Grant 318927 (6Genesis Flagship) and in part by the European Research Council (ERC) Project Theoretical Foundations of Memory Micro-Insertions in Wireless Communications (DUALITY) under Grant 725929. An earlier version of this article was presented in part at the IEEE International Conference on Communications (ICC), June 2020. The associate editor coordinating the review of this article and approving it for publication was W. Chen. *(Corresponding author: MohammadJavad Salehi.)*

MohammadJavad Salehi and Antti Tölli are with the Center for Wireless Communications (CWC), University of Oulu, 90570 Oulu, Finland (e-mail: mohammadjavad.salehi@oulu.fi; antti.tolli@oulu.fi).

Emanuele Parrinello and Petros Elia are with the Communication Systems Department, EURECOM, 06410 Sophia Antipolis, France (e-mail: emanuele.parrinello@eurecom.fi; petros.elia@eurecom.fr).

Seyed Pooya Shariatpanahi is with the School of Electrical and Computer Engineering, College of Engineering, University of Tehran, Tehran 1439957131, Iran (e-mail: p.shariatpanahi@ut.ac.ir).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TWC.2021.3119772.

Digital Object Identifier 10.1109/TWC.2021.3119772

fundamental and generally unavoidable (cf. [7], [8]), a recent new approach in [9] has shown that these limitations are not fundamental in the multi-antenna setting. As we now know from [9], activating multiple transmit antennas can effectively decompose the cache-aided network in a manner that dramatically alleviates the subpacketization bottleneck, thus strongly boosting the true (subpacketization constrained) performance. This performance boost, as well as the ability to achieve the theoretical promises of multi-antenna coded caching with exceedingly small subpacketization complexity, offers a powerful motivation for meaningfully combining coded caching with multi-antenna communications.

It is the case though that, to date, this decomposition principle as it was developed in [9], applies with full optimality only in networks with essentially a modest number of transmit antennas, and in particular, only in networks where the spatial multiplexing gain (which can go up to L) does not exceed the coded caching gain t. In the last two years, extending this optimality to networks with 'larger' antenna arrays has been a known open problem, which is indeed motivated by the upcoming prevalence of very large antenna arrays.

We present a new multi-antenna coded caching structure that resolves this theoretical problem; it achieves the exact optimal one-shot linear DoF $L+t$ even when $L \geq t$ and does so with ultra-low subpacketization. Perhaps more importantly, our scheme, which employs a novel cyclic structure, additionally allows for an ultra-efficient implementation of optimized precoders, thus yielding excellent performance in the low- and mid-SNR range for even large networks.

A. Prior Work

1) Single- and Multi-Antenna Coded Caching: The original coded caching scenario in [2] considers a single-antenna transmitter that communicates to K cache-aided receiving users via a single-stream, normalized-capacity, symmetric broadcast channel. The transmitter hosts a library of N equal-sized files, and each user has a cache memory of size equal to the size of M files. The setting considers two distinct phases: the cache placement phase, which occurs during the off-peak hours, and the content delivery phase, which occurs during the peak hours. The first phase allows for the users' cache memories to be filled with library content, and the second phase employs basic cache-aided multicasting in order to serve $t+1$ users at a time, thus yielding a DoF of $t+1$, where $t := \frac{KM}{N}$. Decoding is achieved by having each user exploit their locally cached content in order to remove the t undesired messages from the received signals. This scheme was proven to be exactly optimal under the constraint of uncoded cache placement [10], [11] and optimal within a gap of two in the general case [12]. Soon after [2], the work in [13] (see also [14], [15]) proposed a decentralized version of the scheme, which, in order to provide good gains, required the file sizes to be very large. In contrast, the work in [16] proposed a single-antenna decentralized scheme for finite-size files, but with significantly worse performance than [2], [13]. Apart from the decentralized setting, the use of coded multicasting was extended to other scenarios such as hierarchical, device-to-device (D2D), and fog radio access networks [17]–[19]. One of the most important scenarios is the multi-antenna/multi-transmitter case, for which the scheme in the aforementioned work in [4] achieves a sum-DoF of $t+L$, which was later proved in [20] to be optimal within a factor of two among all linear one-shot schemes (this gap was then tightened in [6]). The cache-aided interference channel studied in [20] was later extended to cellular networks [21] and heterogeneous parallel channels with\without CSIT [22]. Unlike [20], the work in [23] characterized the global sum-DoF of the cache-aided interference channel (without any restriction on the type of schemes) by proposing a method based on interference alignment techniques that achieves an approximately optimal DoF. In the specific case where each transmitter can cache the entire library (coinciding with the multi-antenna coded caching problem), the scheme in [23] achieves a larger sum-DoF than the optimal linear single-shot sum-DoF. However, the complexity of interference alignment makes the scheme very far from practical, which contrasts with our paper that aims to provide a practical scheme with low complexity. In this regard, we consider only one-shot linear schemes throughout this paper, and by optimal sum-DoF, we mean optimal achievable sum-DoF among all such schemes.

2) Multi-Antenna Coded Caching in the Finite-SNR Regime: The implementation of coded caching techniques in multi-antenna wireless networks initially emphasized the (high-SNR) DoF setting and thus focused on using basic zero-forcing (ZF) precoders [5], [24]. Subsequently, the emphasis was shifted on the lower SNR regimes, with the work in [25] replacing ZF precoders with optimized beamformers that introduced the capability of controlling, rather than completely nulling out, the inter-stream interference. The subsequent work in [26] emphasized both the performance and complexity of beamformer designs, introducing novel optimized precoders that properly controlled the multiplexing gain and the size of the corresponding multiple-access channel (MAC) that is experienced during decoding. In particular, it was shown how operating at a multiplexing gain that is smaller than L can reduce beamformer design complexity as well as yield higher beamforming gains, which are crucial in the low-SNR regime. Controlling the spatial multiplexing gain is also considered in [27], where numerical simulations are used to find the best multiplexing gain for various network parameters, including the coded caching gain. Taking another point of view, the work in [28] proposed a new scheme that limits the number of messages received by a user in each time slot, in order to reduce complexity while maintaining satisfactory rates. Similar works include [29]–[31]. While numerical evaluations suggest that the aforementioned schemes can perform well, this performance is limited to very small network scenarios, mainly due to their massive subpacketization requirements.

3) Subpacketization Bottleneck: To date, in the single-antenna setting, any high-performance coded caching scheme requires a subpacketization that grows — for fixed $\frac{M}{N}$ — exponentially or near-exponentially with K. The scheme in [2] requires subpacketization $\binom{K}{t}$, and as we know from [7], decentralized schemes (cf. [13]) also require exponential subpacketization in order to achieve linear

caching gains. Along similar lines, we know from [32] that under basic assumptions, there exists no single-antenna coded caching scheme that enjoys both linear caching gains and linear subpacketization. Nevertheless, several works in the literature have tried to reduce the required subpacketization in the shared-link coded caching problem. Notably, in [8], Placement Delivery Array (PDA) is proposed as a systematic approach for reducing subpacketization in centralized schemes. For a very large number of users K, in [33], a hypergraph-based scheme is introduced to trade-off subpacketization with performance, and in [34], it is shown that linearly scaling gains (with K) are achievable with a subpacketization that also scales almost linearly. A more recent effort was made in [35], where line graphs and projective geometry are used to achieve an asymptotically (as $K \to \infty$) constant performance with a sub-exponential subpacketization as long as the users' caches are sufficiently large. In the same line of research, other interesting works are [36]–[38]. However, despite the large literature on the topic, no scheme is known to achieve good gains with a relatively small subpacketization and practical values for system parameters.

In the context of multi-antenna coded caching, the situation is different. While the original scheme in [5] required an astronomical subpacketization of $\binom{K}{t}\binom{K-t-1}{L-1}$, the recent work in [9] showed that if $\frac{K}{L}$ and $\frac{t}{L}$ are both integers, the optimal DoF $t+L$ is achievable with a subpacketization of $\binom{K/L}{t/L}$, which is dramatically less than the subpacketization in [2], [5]. This directly means that under fixed subpacketization constraints (fixed file size), adding multiple antennas can multiplicatively boost the real (subpacketization-constrained) DoF by a factor of L. The main idea behind the work in [9] is to employ basic user-grouping techniques to endow groups of users with the same cache content and then apply a specific precoding approach that decomposes the network of users into effectively parallel coded caching problems. While, as we know from [39], for single-antenna setups, having shared caches between the users causes an inevitable DoF loss, the work in [9] has proven that multi-antenna shared-cache setups need not suffer from DoF losses. Of course, this is valid under the assumption that $\frac{K}{L}$ and $\frac{t}{L}$ are integers.[1]

Another interesting work can be found in [40], which proposes a DoF-optimal scheme that yields a reduction in transmission and decoding complexity compared to the optimized beamformer scheme of [26], albeit with a small reduction in performance compared to [26], and also with an exponential subpacketization $\binom{K}{t}$. In another line of work, [41] provides a novel algorithm that reduces the channel state information (CSI) requirements, and does so with subpacketization $L_c\binom{K_c}{t}$, where $L_c := \frac{L+t}{t+1}$ and $K_c := \frac{K}{L_c}$. Finally, [42], [43] explore, under the assumption of $K = t+L$, how subpacketization can be traded-off with performance. To date, existing multi-antenna schemes either exhibit subpacketization requirements that are exponential in K, or do not experience DoF optimality in scenarios where $L > t$.

[1]The scheme of [9] suffers DoF losses (and also increased subpacketization) if either $\frac{K}{L}$ or $\frac{t}{L}$ is non-integer. The DoF is reduced by a multiplicative factor that can reach 2 when $L > t$, and can reach $\frac{3}{2}$ when $L < t$.

B. Our Contribution

Motivated by the ever-increasing sizes of antenna arrays, we proceed to bridge the aforementioned gap and provide a very low-subpacketization coded caching algorithm that achieves the optimal one-shot linear DoF of $L+t$ even in networks with 'larger' antenna arrays such that $L \geq t$. Subsequently, knowing well that in the low-to-moderate SNR regimes beamforming gains can be as important as multiplexing gains, we proceed to consider the more general scenario where the multiplexing gain $\alpha \leq L$ is traded off with an ability to beamform in a manner that compensates the well-known effects of the worst-user channel condition. In our case, the multiplexing gain $\alpha \geq t$ is treated as a design parameter calibrated not only for yielding excellent finite-SNR performance but also for fine-tuning subpacketization and complexity requirements.

The proposed multi-antenna *cyclic-caching scheme* enjoys a novel structure that attains the chosen sum-DoF of $t+\alpha$, as well as the holy trinity of high beamforming gains, reduced subpacketization, and reduced beamforming complexity. In particular, the scheme requires an ultra-low subpacketization $\frac{K(t+\alpha)}{\phi^2_{K,t,\alpha}}$, where $\phi_{K,t,\alpha} = gcd(K,t,\alpha)$; this is currently the smallest known subpacketization out of all one-shot linear schemes with optimal sum-DoF. Most importantly, our cyclic caching method also eliminates the requirement of multicasting, thus interestingly enabling optimized beamformers to be designed with massively reduced computational complexity. These optimized beamformers allow for an interplay between cache-based cancellation, nulling out, and controlling the interference, resulting in a very considerable performance boost in low- and mid-SNR communications. In the end, our work provides a method for a high-performance hyper-efficient use of coded caching in large multi-antenna networks and for any SNR value.

C. Structure and Notation

We use $[K]$ to denote the set $\{1,2,\ldots,K\}$ and $[i:j]$ to represent the vector $[i \;\; i+1 \;\; \ldots \;\; j]$. Boldface upper- and lower-case letters denote matrices and vectors, respectively. $\mathbf{V}[i,j]$ refers to the element at the i-th row and j-th column of matrix $\mathbf{V}$, and $\mathbf{w}[i]$ represents the i-th element in vector $\mathbf{w}$. Moreover, $\mathbf{w} = [\mathbf{u};\mathbf{v}]$ refers to a vector $\mathbf{w}$ that is formed by concatenating vectors $\mathbf{u}$ and $\mathbf{v}$. Sets are denoted by calligraphic letters. For two sets $\mathcal{A}$ and $\mathcal{B}$, $\mathcal{A}\backslash\mathcal{B}$ refers to the set of elements in $\mathcal{A}$ that are not in $\mathcal{B}$, and $|\mathcal{A}|$ is the number of elements in $\mathcal{A}$.

The rest of the paper is organized as follows. Section II presents the system model, while Section III presents the new cyclic caching scheme. In this section, we show that for any $\alpha \geq t$, a sum-DoF of $t+\alpha$ is possible, first with subpacketization $K(t+\alpha)$, and then with a subpacketization that is further reduced by a factor of $\phi^2_{K,t,\alpha}$. Section IV presents complexity analysis and performance simulations, while Section V concludes the paper.

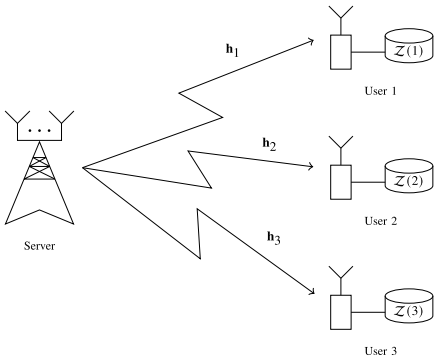


Fig. 1. Illustration of the communication setup for a network of $K = 3$ users.

II. System Model

A. Network Setup

We consider a multiple-input, single-output (MISO) broadcast setting, where a single server, equipped with L transmit antennas, communicates with K single-antenna receiving users over a shared wireless link. An illustration of the considered communication setup for a small network of $K = 3$ users is provided in Figure 1. The server has access to a library $\mathcal{F}$ of $N \geq K$ files, where each file $W \in \mathcal{F}$ has a size of f bits. We assume that every user has a cache memory of size Mf bits. As mentioned earlier, we use $t := K\frac{Mf}{Nf}$ to denote the total cache size in the network normalized by the size of the library. In essence, t — which we will assume to be an integer — indicates the redundancy with which the library can be stored across the network.

During the cache placement phase, the placement algorithm operates without any prior knowledge of future requests. We use $\mathcal{Z}(k)$ to denote the cache contents of user $k \in [K]$ after the placement phase is completed. At the beginning of the delivery phase, each user $k \in [K]$ reveals its requested file $W(k) \in \mathcal{F}$ to the server. After receiving the demand set $\mathcal{D} = \{W(k) \mid k \in [K]\}$, the server follows the delivery algorithm to transmit the requested subpackets to the users. This will involve the transmission of some I transmission vectors $\{\mathbf{x}_i\} \in \mathbb{C}^L$, $i \in [I]$, where I is given by the delivery algorithm. These transmission vectors are transmitted in consecutive time intervals or separate frequency bins,[2] using the array of L antennas. After $\mathbf{x}_i$ is transmitted, user k receives $y_i(k) = \mathbf{h}_k^H \mathbf{x}_i + w_i(k)$, where $\mathbf{h}_k \in \mathbb{C}^L$ denotes the channel vector and $w_i(k) \sim \mathbb{CN}(0, N_0)$ denotes the observed noise at user k. Furthermore, we consider a slow-fading model in which the channel vectors remain constant during each time interval i, and we assume that full channel state information (CSI) is available at the server.[3]

[2]For comparison with other schemes, we will generally assume that transmissions here are made in consecutive time intervals.

[3]Improving CSI accuracy is a well-studied topic in the literature. In general, the designs are specific to the underlying uncertainty model, i.e., whether CSI error is bounded [44] or unbounded [45]. If the error is bounded, robust (worst-case) beamforming solutions can be computed that guarantee the achievability of the max-min SINR for all possible realizations of channel uncertainty. For unbounded error, statistically robust solutions can be provided if the estimation noise is assumed to follow a known distribution. In such a scenario, fixed-point iterations are used to provide a robust solution by adding the noise contribution from CSI uncertainty to the thermal noise.

Let $\mathcal{X}_i \subseteq [K]$ denote the set of users targeted by $\mathbf{x}_i$, and let T_i denote the duration of time interval i required so that every user in $\mathcal{X}_i$ decodes its intended data from $\mathbf{x}_i$. If we consider L_i to be the length of the codeword transmitted at time slot i, and if we define R_i to be the multicast rate at which the server transmits a common message to all users in $\mathcal{X}_i$, then T_i is simply the ratio between L_i and R_i. We will use the metric of the *symmetric rate*, which describes the total number of bits per second with which each user is served. Notably, we will consider the worst-case metric, corresponding to the symmetric rate at which the system can serve all users in the network irrespective of the demand set $\mathcal{D}$. Given that the delivery phase has an overall duration of $\sum_{i=1}^{I} T_i$, and given that there are K users, the symmetric rate can be computed as

$$R_{sym} = \frac{Kf}{\sum_{i=1}^{I} T_i} = \frac{Kf}{\sum_{i=1}^{I} \frac{L_i}{R_i}}. \tag{1}$$

Our aim is to design a placement and delivery scheme that maximizes R_{sym}.

B. Building the Transmission Vectors

We use linear precoding to build the transmission vectors. A generic transmission vector $\mathbf{x}_i$ is built as

$$\mathbf{x}_i = \sum_{k \in \mathcal{X}_i} \mathbf{w}_i(k) X_i(k), \tag{2}$$

where $X_i(k)$ is the data codeword transmitted to user k, and $\mathbf{w}_i(k) \in \mathbb{C}^L$ is the beamforming vector used for $X_i(k)$. The beamforming vectors $\mathbf{w}_i(k)$ are here designed to maximize the worst-user rate, or equivalently, the worst user's SINR (signal to interference and noise ratio), as $\mathbf{x}_i$ is transmitted. Thus, given the transmission model in (2), the multicast rate at time interval i is calculated as

$$R_i = \log(1 + \gamma_i^*), \tag{3}$$

in which γ_i^* is defined as

$$\gamma_i^* = \max_{\mathbf{w}_i(k)} \min_{k \in \mathcal{X}_i} \text{SINR}_k \quad s.t. \quad \sum_{k \in \mathcal{X}_i} ||\mathbf{w}_i(k)||^2 \leq P_T, \tag{4}$$

where SINR_k is the received SINR at user k and P_T is the available transmission power. We discuss this optimization problem in more details in the next section.

As suggested before, we will consider that each transmission vector serves $|\mathcal{X}_i| = t + \alpha$ users, where $\alpha \leq L$ is the multiplexing gain and is treated as a parameter of choice that can be tuned to obtain a better rate performance at finite-SNR. This α represents the number of independent streams in each transmission, and hence, reducing it implies sacrificing some spatial multiplexing gain for the purpose of increasing the beamforming gain, which can help reduce the worst-user effect in the finite-SNR regime. As we discuss later on, this same α can also be calibrated to control subpacketization and beamformer design complexity.

III. Cyclic Caching for Reduced Subpacketization

In this section, we present our low-complexity high performance cyclic caching scheme, which can be applied to any MISO setup in which $\alpha \geq t$.[4] The following theorem summarizes the DoF and subpacketization performance of the scheme:

Theorem 1: For the large MISO broadcast setup with $t \leq \alpha \leq L$, the sum DoF of $t+\alpha$ is achievable with a subpacketization

$$\frac{K(t+\alpha)}{\big(gcd(K,t,\alpha)\big)^2}. \tag{5}$$

Proof: The proof is found in this current section, where we present the designed cyclic caching scheme and show that it employs the above subpacketization to achieve the sum DoF of $t+\alpha$. ∎

In what follows, we first introduce a cache placement algorithm in Section III-A, which is based on a well-defined placement matrix and requires each file to be split into $K(t+\alpha)$ smaller parts (subpackets). In Sections III-B and III-C, we explain the delivery phase, in which the missing data parts are delivered to the requesting users with $I = K(K-t)$ multicast transmission vectors, each serving $t+\alpha$ subpackets to $t+\alpha$ different users. In Section III-D, using an example network, we show that this delivery algorithm follows a simple graphical representation that involves circular shifts of two vectors over a tabular structure. Overall, in Sections III-A to III-D, we present a scheme that satisfies all the requests with multicast transmissions that always contain $t+\alpha$ subpackets, implying a DoF of $t+\alpha$ with subpacketization $K(t+\alpha)$. Finally, in Section III-F we show that by properly applying a user-grouping technique, subpacketization is further reduced by a factor of $\big(gcd(K,t,\alpha)\big)^2$, without any DoF loss.

Remark 1: When $\alpha = L$, the achieved DoF $t+L$ is exactly optimal under the assumption of one-shot linear schemes and uncoded placement (cf. [6]).

We note that, for fixed t and L, the above integer subpacketization scales linearly with K. This allows applying coded caching in larger networks, and entails the benefit of a reduced number of necessary transmissions which in turn implies a reduced number of beamformer design problems that need to be solved. As a quick comparison, if $K = 20, t = 4, L = \alpha = 8$, the proposed scheme requires subpacketization of 15, while the schemes in [40] and [5] respectively require (approximately) 5×10^3 and 3×10^7 subpackets. More comparisons are provided in Section IV.

A. Cache Placement

For cache placement, we use a $K \times K$ binary placement matrix $\mathbf{V}$ where the first row has t consecutive 1's (other elements are zero) and each subsequent row is a circular shift of the previous row by one column. Given $\mathbf{V}$, we split each file W into K packets W_p, $p \in [K]$, and each packet W_p into $t+\alpha$ smaller subpackets W_p^q. Then for every $p, k \in [K]$, if $\mathbf{V}[p,k] = 1$, W_p^q is stored in the cache memory of user k, $\forall W \in \mathcal{F}, q \in [t+\alpha]$.[5]

Example 1: For a scenario of $K = 6$, $t = 2$, $\alpha = 3$, $\mathbf{V}$ is built as

$$\mathbf{V} = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \tag{6}$$

and the resulting subpacketization is $K \times (t+\alpha) = 6 \times (2+3) = 30$. For example, the cache contents of users 1 and 2 can be found from (6) as

$$\mathcal{Z}(1) = \{W_1^q, W_6^q \,;\, \forall W \in \mathcal{F},\, q \in [5]\},$$
$$\mathcal{Z}(2) = \{W_1^q, W_2^q \,;\, \forall W \in \mathcal{F},\, q \in [5]\}.$$

The cache contents of users $3-6$ can be written accordingly.

B. Content Delivery

In cyclic caching, the content delivery phase consists of K rounds, where in each round we build $K-t$ transmission vectors. Thus, the content delivery is completed after $I = K(K-t)$ transmissions. We use $\mathbf{x}_j^r$ to denote the transmission vector $j \in [K-t]$ at transmission round $r \in [K]$.[6] By transmitting $\mathbf{x}_j^r$, useful data packets are delivered to a set of $t+\alpha$ users. We define the user index vector $\mathbf{k}_j^r$ to denote the set of users being targeted by $\mathbf{x}_j^r$, and the packet index vector $\mathbf{p}_j^r$ to contain the packet indices targeted for the users in $\mathbf{k}_j^r$. In other words, using $\mathbf{x}_j^r$, we transmit (part of) the packet $W_{\mathbf{p}_j^r[n]}(\mathbf{k}_j^r[n])$ to each user $\mathbf{k}_j^r[n]$, $n = 1, \ldots, t+\alpha$. Both $\mathbf{k}_j^r$ and $\mathbf{p}_j^r$ vectors are built recursively. Let us use $\%$ sign to denote the mod operator with an offset of one. It is defined as

$$a\%b = ((a-1) \bmod b) + 1, \tag{7}$$

such that $a\%a = a$ and $(a+b)\%a = b\%a$. Then, $\mathbf{k}_j^1$ and $\mathbf{p}_j^1$ are built as

$$\mathbf{k}_j^1 = \Big[\, [1:t] \,;\, \big(([1:\alpha] + j - 1)\%(K-t)\big) + t \,\Big],$$
$$\mathbf{p}_j^1 = \Big[\big((t + j - [1:t])\%(K-t)\big) + [1:t] \,;\, \mathbf{e}(\alpha)\Big], \tag{8}$$

where $\mathbf{e}(m)$ is a vector of 1's with size m (e.g., $\mathbf{e}(3) = [1\ 1\ 1]$). For the next transmission rounds, i.e., $1 < r \leq K$, we simply build $\mathbf{k}_j^r$ and $\mathbf{p}_j^r$, using $\mathbf{k}_j^1$ and $\mathbf{p}_j^1$, as

$$\mathbf{k}_j^r = \big(\mathbf{k}_j^1 + r\big)\%K, \quad \mathbf{p}_j^r = \big(\mathbf{p}_j^1 + r\big)\%K. \tag{9}$$

To gain a better insight into how $\mathbf{k}_j^r$ and $\mathbf{p}_j^r$ are built, in Section III-D, we offer a simple graphical representation, which is based on circular shift operations over a tabular structure. In the following, we provide $\mathbf{k}_j^r$ and $\mathbf{p}_j^r$ vectors for the small network scenario given in Example 1.

[4]For setups with $t > \alpha$, one can use the coded caching scheme presented in [9] for reduced subpacketization.

[5]The placement matrix $\mathbf{V}$ used in this paper is a special case of valid placement matrices introduced in [42].

[6]In the general transmission vector model (2), $\mathbf{x}_j^r$ corresponds to $\mathbf{x}_i$, $i = (r-1)(K-t) + j$.

Example 2: In the scenario of Example 1, content delivery consists of six rounds, where at each round four transmission vectors are built. The user and packet index vectors for the first and second rounds are given as

$$\begin{aligned}
&\mathbf{k}_1^1 = [1\ 2\ 3\ 4\ 5], \quad \mathbf{p}_1^1 = [3\ 3\ 1\ 1\ 1],\\
&\mathbf{k}_2^1 = [1\ 2\ 4\ 5\ 6], \quad \mathbf{p}_2^1 = [4\ 4\ 1\ 1\ 1],\\
&\mathbf{k}_3^1 = [1\ 2\ 5\ 6\ 3], \quad \mathbf{p}_3^1 = [5\ 5\ 1\ 1\ 1],\\
&\mathbf{k}_4^1 = [1\ 2\ 6\ 3\ 4], \quad \mathbf{p}_4^1 = [2\ 6\ 1\ 1\ 1],
\end{aligned} \tag{10}$$

and

$$\begin{aligned}
&\mathbf{k}_1^2 = [2\ 3\ 4\ 5\ 6], \quad \mathbf{p}_1^2 = [4\ 4\ 2\ 2\ 2],\\
&\mathbf{k}_2^2 = [2\ 3\ 5\ 6\ 1], \quad \mathbf{p}_2^2 = [5\ 5\ 2\ 2\ 2],\\
&\mathbf{k}_3^2 = [2\ 3\ 6\ 1\ 4], \quad \mathbf{p}_3^2 = [6\ 6\ 2\ 2\ 2],\\
&\mathbf{k}_4^2 = [2\ 3\ 1\ 4\ 5], \quad \mathbf{p}_4^2 = [3\ 1\ 2\ 2\ 2],
\end{aligned} \tag{11}$$

respectively. The user and packet index vectors for the other rounds are built similarly.

Two other variables are needed to build the transmission vector $\mathbf{x}_j^r$. First, we introduce the subpacket index $q(W,p)$, where $W \in \mathcal{F}$ denotes a general file and $p \in [K]$ is the packet index. The subpacket index $q(W,p)$ indicates which subpacket of W_p should be transmitted, the next time it is included in a transmission vector. For every $W \in \mathcal{F}$ and $p \in [K]$, $q(W,p)$ is initialized to one, and incremented every time W_p is included in a transmission vector. For notational simplicity, here we use

$$q_j^r(n) := q(W(\mathbf{k}_j^r[n]), \mathbf{p}_j^r[n]). \tag{12}$$

Second, we define the *interference indicator* set $\mathcal{R}_j^r(n)$ as the set of users at which $W_{\mathbf{p}_j^r[n]}^{q_j^r(n)}(\mathbf{k}_j^r[n])$ should be suppressed by beamforming.[7] $\mathcal{R}_j^r(n)$ has $\alpha - 1$ elements and is built as

$$\mathcal{R}_j^r(n) = \Big\{ k \in \mathbf{k}_j^r \backslash \mathbf{k}_j^r[n] \quad | \quad \mathbf{V}\big[\mathbf{p}_j^r[n], k\big] = 0 \Big\}. \tag{13}$$

Finally, the transmission vectors are built as:[8]

$$\mathbf{x}_j^r = \sum_{n=1}^{t+\alpha} \mathbf{w}_{\mathcal{R}_j^r(n)} W_{\mathbf{p}_j^r[n]}^{q_j^r(n)}(\mathbf{k}_j^r[n]). \tag{14}$$

Example 3: For the network considered in Examples 1 and 2, the interference indicator sets for the first transmission round are built as

$$\begin{aligned}
&\mathcal{R}_1^1(1) = \{2,5\}, \quad \mathcal{R}_1^1(2) = \{1,5\}, \quad \mathcal{R}_1^1(3) = \{4,5\},\\
&\mathcal{R}_1^1(4) = \{3,5\}, \quad \mathcal{R}_1^1(5) = \{4,4\}.
\end{aligned} \tag{15}$$

[7]If zero-forcing beamformers are used, $\mathcal{R}_j^r(n)$ denotes the set of users at which $W_{\mathbf{p}_j^r[n]}^{q_j^r(n)}(\mathbf{k}_j^r[n])$ should be nulled-out by beamforming.

[8]The general transmission vector model in (2) is equivalent to (14) via the following index mapping:

$$\begin{aligned}
&k \to \mathbf{k}_j^r[n], \quad \mathbf{w}_i \to \mathbf{w}_{\mathcal{R}_j^r(n)},\\
&\mathcal{X}_i \to \bigcup_{n\in[t+\alpha]} \{\mathbf{k}_j^r[n]\}, \quad X_i(k) \to W_{\mathbf{p}_j^r[n]}^{q_j^r(n)}(\mathbf{k}_j^r[n]).
\end{aligned}$$

C. Decoding at the Receiver

During time interval i, every user $k \in \mathcal{X}_i$ receives

$$y_i(k) = \mathbf{h}_k^H \mathbf{w}_i(k) X_i(k) + \sum_{\hat{k}\in\mathcal{X}_i\backslash\{k\}} \mathbf{h}_k^H \mathbf{w}_i(\hat{k}) X_i(\hat{k}) + w_i(k), \tag{16}$$

where the first term is the intended codeword and the latter two terms indicate the interference and noise, respectively. Assume $\hat{k} := \mathbf{k}_{\hat{j}}^{\hat{r}}[\hat{n}]$, for some $\hat{j}, \hat{r}, \hat{n}$. Defining $p(\hat{k}) := \mathbf{p}_{\hat{j}}^{\hat{r}}[\hat{n}]$, for every element in the interference term only one of the following options is possible:

1) $\mathbf{V}[p(\hat{k}), k] = 1$ indicates $X_i(\hat{k})$ is in the cache memory of user k, and hence, $\mathbf{h}_k^H \mathbf{w}_i(\hat{k}) X_i(\hat{k})$ can be reconstructed and removed from $y_i(k)$;
2) $\mathbf{V}[p(\hat{k}), k] = 0$ indicates that $\hat{k}$ is in the interference indicator set associated with $X_i(k)$ as defined in (13), and hence, $X_i(\hat{k})$ is suppressed at user k by transmit beamforming.

In both cases, the interference due to $X_i(\hat{k})$ can be controlled and/or completely removed at user k. Since $|\mathcal{X}_i| = t + \alpha$, the proposed scheme allows for serving $t+\alpha$ users in parallel during each transmission interval. The following example clarifies the decoding procedure for a single transmission in a small network. A more detailed explanation is provided in Appendix V-A.

Example 4: Consider the network in Example 1, for which the user and packet index vectors are provided in Example 2 and the interference indicator sets are presented in Example 3. Let us assume the demand set is $\mathcal{D} = \{A, B, C, D, E, F\}$. Then, following (14)*, the first transmission vector in the first round is built as*

$$\mathbf{x}_1^1 = \ \mathbf{w}_{2,5} A_3^1 + \mathbf{w}_{1,5} B_3^1 + \mathbf{w}_{4,5} C_1^1 + \mathbf{w}_{3,5} D_1^1 + \mathbf{w}_{3,4} E_1^1, \tag{17}$$

where the brackets of the interference indicator sets are dropped for notation simplicity. After $\mathbf{x}_1^1$ is transmitted, user 1 receives

$$\begin{aligned}
y_1^1(1) = \mathbf{h}_1^H \mathbf{w}_{2,5} A_3^1 + \underline{\underline{\mathbf{h}_1^H \mathbf{w}_{1,5} B_3^1}} + \underline{\mathbf{h}_1^H \mathbf{w}_{4,5} C_1^1}\\
+ \underline{\mathbf{h}_1^H \mathbf{w}_{3,5} D_1^1} + \underline{\mathbf{h}_1^H \mathbf{w}_{3,4} E_1^1} + w_1^1(1),
\end{aligned} \tag{18}$$

where the single- and double-underlined terms indicate the interference. From Example 1, C_1^1, D_1^1, and E_1^1 are available in the cache memory of user 1, and hence, all the single-underlined terms can be reconstructed and removed from the received signal. On the other hand, following the definition of the interference indicator sets, the double-underlined term (containing B_3^1) is also suppressed at user 1 with the help of the beamforming vectors. As a result, user 1 can decode A_3^1 with controlled interference. Similarly, users $2-5$ can decode B_3^1, C_1^1, D_1^1 and E_1^1, respectively. In Table I, we have summarized how different users decode $\mathbf{x}_1^1$ and extract their requested data.

D. A Graphical Example

For further clarification, we describe the operation of the cyclic caching scheme for the network setup in Example 1,

TABLE I

DECODING PROCESS FOR $\mathbf{x}_1^1$ AT DIFFERENT NETWORK USERS, FOR EXAMPLE 4

Transmission vector: $\mathbf{x}_1^1 = \mathbf{w}_{2,5}A_3^1 + \mathbf{w}_{1,5}B_3^1 + \mathbf{w}_{4,5}C_1^1 + \mathbf{w}_{3,5}D_1^1 + \mathbf{w}_{3,4}E_1^1$

User	Available in cache	Supp. by beamformer	Useful data	SINR
1	C_1^1, D_1^1, E_1^1	B_3^1	A_3^1	$\frac{\lvert\mathbf{h}_1^H\mathbf{w}_{2,5}\rvert^2}{\lvert\mathbf{h}_1^H\mathbf{w}_{1,5}\rvert^2+N_0}$
2	C_1^1, D_1^1, E_1^1	A_3^1	B_3^1	$\frac{\lvert\mathbf{h}_1{}^2{}^H\mathbf{w}_{1,5}\rvert^2}{\lvert\mathbf{h}_2^H\mathbf{w}_{2,5}\rvert^2+N_0}$
3	A_3^1, B_3^1	D_1^1, E_1^1	C_1^1	$\frac{\lvert\mathbf{h}_3^H\mathbf{w}_{4,5}\rvert^2}{\lvert\mathbf{h}_3^H\mathbf{w}_{3,5}\rvert^2+\lvert\mathbf{h}_3^H\mathbf{w}_{3,4}\rvert^2+N_0}$
4	A_3^1, B_3^1	C_1^1, E_1^1	D_1^1	$\frac{\lvert\mathbf{h}_4^H\mathbf{w}_{3,5}\rvert^2}{\lvert\mathbf{h}_4^H\mathbf{w}_{4,5}\rvert^2+\lvert\mathbf{h}_4^H\mathbf{w}_{3,4}\rvert^2+N_0}$
5	A_3^1, B_3^1	C_1^1, D_1^1	E_1^1	$\frac{\lvert\mathbf{h}_5^H\mathbf{w}_{3,4}\rvert^2}{\lvert\mathbf{h}_5^H\mathbf{w}_{4,5}\rvert^2+\lvert\mathbf{h}_5^H\mathbf{w}_{3,5}\rvert^2+N_0}$
6	–	–	–	–

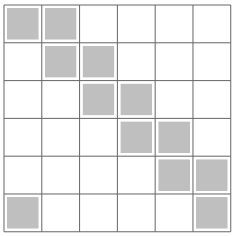


Fig. 2. Graphical illustration for Example 1.

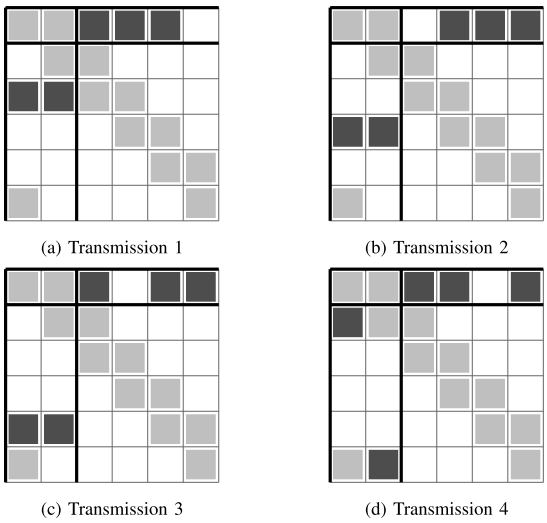
using a graphical representation of the placement matrix $\mathbf{V}$ in Figure 2. In this figure, each column represents a user, and each row denotes a packet index. For example, the first column represents user one, and the first row stands for the first packet of all files, i.e., W_1^q, $\forall W \in \mathcal{F}, q \in [t+\alpha]$. Lightly shaded entries indicate packets that are cached at the user. For example, W_1^q, W_6^q are stored at user 1, $\forall W \in \mathcal{F}, q \in [t+\alpha]$.

In the subsequent figures, we use darkly shaded entries to indicate which packet indices of the requested files are sent during each transmission. The column and row indices of these darkly shaded entries are extracted from the user and packet index vectors. For our example network, the user and packet index vectors for the first and second transmission rounds are provided in (10) and (11), and their graphical representations are depicted in Figures 3 and 4. For example, Fig. 3a corresponds to the first transmission of the first round, where users $\mathbf{k}_1^1 = [1,2,3,4,5]$ receive subpackets of packets indicated by $\mathbf{p}_1^1 = [3,3,1,1,1]$. For simplicity, let us assume the demand set is $\mathcal{D} = \{A, B, C, D, E, F\}$. Then, users 1-5 receive A_3^1, B_3^1, C_1^1, D_1^1 and E_1^1, respectively.

The transmission vectors can be easily reconstructed using the graphical representations. For example, Fig. 3 implies that the following transmission vectors are generated in the first round:

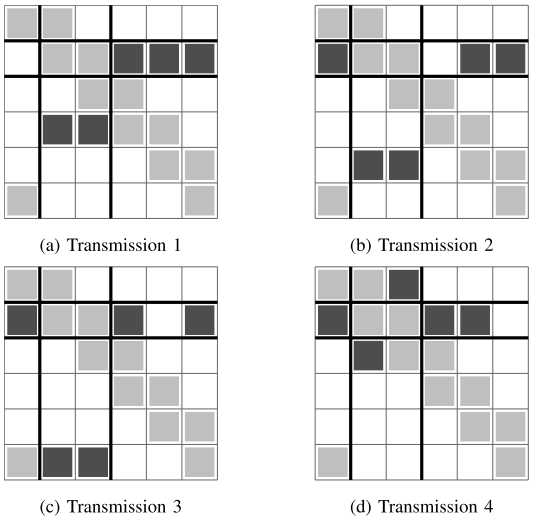
$$\begin{aligned}
\mathbf{x}_1^1 &= \mathbf{w}_{2,5}A_3^1 + \mathbf{w}_{1,5}B_3^1 + \mathbf{w}_{4,5}C_1^1 + \mathbf{w}_{3,5}D_1^1 + \mathbf{w}_{3,4}E_1^1,\\
\mathbf{x}_2^1 &= \mathbf{w}_{2,6}A_4^1 + \mathbf{w}_{1,6}B_4^1 + \mathbf{w}_{5,6}D_1^2 + \mathbf{w}_{4,6}E_1^2 + \mathbf{w}_{4,5}F_1^1,\\
\mathbf{x}_3^1 &= \mathbf{w}_{2,3}A_5^1 + \mathbf{w}_{1,3}B_5^1 + \mathbf{w}_{6,3}E_1^3 + \mathbf{w}_{5,3}F_1^2 + \mathbf{w}_{5,6}C_1^2,\\
\mathbf{x}_4^1 &= \mathbf{w}_{4,6}A_2^1 + \mathbf{w}_{3,4}B_6^1 + \mathbf{w}_{3,4}F_1^3 + \mathbf{w}_{6,4}C_1^3 + \mathbf{w}_{6,3}D_1^3,
\end{aligned}$$

where the brackets of the interference indicator sets are dropped for notation simplicity. Note that according to (12),



(a) Transmission 1 (b) Transmission 2 (c) Transmission 3 (d) Transmission 4

Fig. 3. Graphical illustration of the first round $r = 1$.



(a) Transmission 1 (b) Transmission 2 (c) Transmission 3 (d) Transmission 4

Fig. 4. Graphical illustration of the second round $r = 2$.

we have $q_j^1(n) \equiv q(W(\mathbf{k}_j^1[n]), \mathbf{p}_j^1[n])$, and hence, $q_j^1(n)$ is incremented every time $W_{\mathbf{p}_j^1[n]}(\mathbf{k}_j^1[n])$ appears in a transmission vector. As a result, the subpacket index for the packet C_1 is incremented from one to three, as it has appeared in $\mathbf{x}_1^1$, $\mathbf{x}_3^1$ and $\mathbf{x}_4^1$, respectively.

Following the same procedure, using Figure 4, for the second round we have

$$\begin{aligned}
\mathbf{x}_1^2 &= \mathbf{w}_{3,6}B_4^2 + \mathbf{w}_{2,6}C_4^1 + \mathbf{w}_{5,6}D_2^1 + \mathbf{w}_{4,6}E_2^1 + \mathbf{w}_{4,5}F_2^1,\\
\mathbf{x}_2^2 &= \mathbf{w}_{3,1}B_5^2 + \mathbf{w}_{2,1}C_5^1 + \mathbf{w}_{6,1}E_2^2 + \mathbf{w}_{5,1}F_2^2 + \mathbf{w}_{5,6}A_2^2,\\
\mathbf{x}_3^2 &= \mathbf{w}_{3,4}B_6^2 + \mathbf{w}_{2,4}C_6^1 + \mathbf{w}_{1,4}F_2^3 + \mathbf{w}_{6,4}A_2^3 + \mathbf{w}_{6,1}D_2^2,\\
\mathbf{x}_4^2 &= \mathbf{w}_{5,1}B_3^2 + \mathbf{w}_{4,5}C_1^4 + \mathbf{w}_{4,5}A_2^4 + \mathbf{w}_{1,5}D_2^3 + \mathbf{w}_{1,4}E_2^3.
\end{aligned}$$

From Figures 3 and 4, it can be seen that the transmissions vectors $\mathbf{x}_2^r$, $\mathbf{x}_3^r$, and $\mathbf{x}_4^r$ in round r are built by circular shifts of the first transmission vector $\mathbf{x}_1^r$ over the non-shaded cells of the tabular grid and in two perpendicular directions. Specifically, the first two terms in $\mathbf{x}_1^r$ are shifted vertically, while the other three are shifted horizontally. This procedure is

highlighted in sub-figures using wider border lines. Moreover, comparing Figures 3 and 4, it is clear that the vectors in the second transmission round are diagonally shifted versions of the vectors in the first round. This property is the intuition behind the cyclic caching name and results from the recursive procedure in (9), where the mod operator is used to build $\mathbf{k}_j^r$ and $\mathbf{p}_j^r$ vectors for $r > 1$.

E. Beamformer Design

As discussed earlier, we use optimized beamformers to build the transmission vectors. These beamformers result in a better rate compared with zero-forcing, especially in the low-SNR regime, as they allow balancing the detrimental impact of noise and the inter-stream interference [26]. However, optimized beamformers may require non-convex optimization problems to be solved (due to interference from unwanted terms), making the problem computationally intractable even for moderate K values. Interestingly, in addition to requiring much-reduced subpacketization, cyclic caching also manages to eliminate the requirement of multicasting, thus enabling optimized beamformers to be designed with much less computational complexity.

As $t+\alpha$ users are served simultaneously by each transmission vector, symmetric rate maximization is equivalent to maximizing the worst user rate (among served users), which, in turn, is equivalent to maximizing the worst user SINR. Naturally, the unwanted terms canceled out using the local cache contents are *not* considered interference in optimized SINR expressions. The optimized beamformer vectors for the j-th transmission in round r can be found by solving the optimization problem

$$\max_{\mathbf{w}_{\mathcal{R}_j^r(n)}} \min_{n\in[t+\alpha]} \frac{|\mathbf{h}_{\mathbf{k}_j^r[n]}^H \mathbf{w}_{\mathcal{R}_j^r(n)}|^2}{\sum_{b:\mathcal{R}_j^r[b]\ni \mathbf{k}_j^r[n]} |\mathbf{h}_{\mathbf{k}_j^r[n]}^H \mathbf{w}_{\mathcal{R}_j^r(b)}|^2 + N_0}$$
$$s.t. \quad \sum_{n\in[t+\alpha]} |\mathbf{w}_{\mathcal{R}_j^r(n)}|^2 \le P_T. \tag{19}$$

Example 5: Consider the network in Example 1 and the transmitted signal vector $\mathbf{x}_1^1$ *in* (17) *for the first transmission of the round* $r = 1$. *The optimized beamformers* $\mathbf{w}_{25}, \mathbf{w}_{15}, \mathbf{w}_{45}, \mathbf{w}_{35}, \mathbf{w}_{34}$ *can be found by solving*

$$\max_{\mathbf{w}_{\mathcal{R}}} \min \Big\{ \frac{|\mathbf{h}_1^H \mathbf{w}_{25}|^2}{\lambda_1 + N_0}, \frac{|\mathbf{h}_2^H \mathbf{w}_{15}|^2}{\lambda_2 + N_0}, \frac{|\mathbf{h}_3^H \mathbf{w}_{45}|^2}{\lambda_3 + N_0}, \frac{|\mathbf{h}_4^H \mathbf{w}_{35}|^2}{\lambda_4 + N_0}, \frac{|\mathbf{h}_5^H \mathbf{w}_{34}|^2}{\lambda_5 + N_0} \Big\}$$
$$s.t. \ \ |\mathbf{w}_{25}|^2 + |\mathbf{w}_{15}|^2 + |\mathbf{w}_{45}|^2 + |\mathbf{w}_{35}|^2 + |\mathbf{w}_{34}|^2 \le P_T,$$

where λ_k *denotes the interference at user* k, *given as*

$$\lambda_1 = |\mathbf{h}_1^H \mathbf{w}_{15}|^2,$$
$$\lambda_2 = |\mathbf{h}_2^H \mathbf{w}_{25}|^2,$$
$$\lambda_3 = |\mathbf{h}_3^H \mathbf{w}_{35}|^2 + |\mathbf{h}_3^H \mathbf{w}_{34}|^2,$$
$$\lambda_4 = |\mathbf{h}_4^H \mathbf{w}_{34}|^2 + |\mathbf{h}_4^H \mathbf{w}_{45}|^2,$$
$$\lambda_5 = |\mathbf{h}_5^H \mathbf{w}_{15}|^2 + |\mathbf{h}_5^H \mathbf{w}_{25}|^2 + |\mathbf{h}_5^H \mathbf{w}_{35}|^2 + |\mathbf{h}_5^H \mathbf{w}_{45}|^2.$$

In cyclic caching, as also demonstrated in Example 5, the number of interfering messages experienced by each user does not need to be the same, in general. For the transmission vector $\mathbf{x}_1^1$ considered in Example 5, users 1-5 experience 1, 1, 2, 2, and 4 interfering messages, respectively. This unevenness is an intrinsic characteristic of the proposed cyclic caching scheme, while each message is still suppressed at exactly $\alpha-1$ users (in Example 5, there exist exactly $\alpha - 1 = 2$ users in each interference indicator set).[9]

The optimized beamformer design problems tend to be non-convex in general and require iterative methods such as successive convex approximation (SCA) to be used [26]. Such methods can be computationally complex and make the implementation infeasible, especially for large networks. However, the unique unicasting nature of the cyclic caching transmission and the max-min SINR objective in (19) make the optimization problem quasi-convex [46], [47], and hence, allow us to use uplink-downlink duality to attain a simple iterative solution. As the beamformer design with uplink-downlink duality is thoroughly discussed in [47], here we only briefly review the required process. Denoting the normalized receive beamforming vectors for the dual uplink channel as $\mathbf{v}_{\mathcal{R}_j^r(n)}$, $r \in [K]$, and $j \in [K-t]$, the uplink-downlink duality necessitates

$$\sum_{n\in[t+\alpha]} \nu_n \|\mathbf{v}_{\mathcal{R}_j^r(n)}\|^2 = \sum_{n\in[t+\alpha]} \|\mathbf{w}_{\mathcal{R}_j^r(n)}\|^2, \tag{20}$$

where the beamforming vectors $\mathbf{v}_{\mathcal{R}_j^r(n)}$ and their power values ν_n can be found by maximizing the minimum of dual uplink SINR expressions:

$$\max_{\mathbf{v}_{\mathcal{R}_j^r(n)}, \nu_n} \min_{n\in[t+\alpha]} \gamma_n = \frac{\nu_n |\mathbf{h}_{\mathbf{k}_j^r[n]}^H \mathbf{v}_{\mathcal{R}_j^r(n)}|^2}{\sum_{b:\mathcal{R}_j^r[b]\ni \mathbf{k}_j^r[n]} \nu_b |\mathbf{h}_{\mathbf{k}_j^r[b]}^H \mathbf{v}_{\mathcal{R}_j^r(n)}|^2 + N_0}$$
$$s.t. \quad \sum_{n\in[t+\alpha]} \nu_n \le P_T, \quad \|\mathbf{v}_{\mathcal{R}_j^r(n)}\|^2 = 1 \ \forall \ n. \tag{21}$$

Note that the interference terms in the denominator of (21) have different indices compared with (19). The dual uplink optimization problem in (21) is quasi-convex and can be solved optimally for the given unicast group [47]. Here we use a standard iterative approach, where we adjust (e.g., by bisection) a target SINR value, denoted by $\bar{\gamma}$, until the power constraint is met with a desired convergence level $|P_T - \sum_{n\in[t+\alpha]} \nu_n| < \epsilon$. However, this requires finding the power coefficients ν_n resulting in the minimum total power $\sum_{n\in[t+\alpha]} \nu_n$, for a given target SINR value $\bar{\gamma}$. We use another internal iterative loop to address this issue. We first note that the (normalized) MMSE receiver $\mathbf{v}_{\mathcal{R}_j^r(n)} = \frac{\bar{\mathbf{v}}}{\|\bar{\mathbf{v}}\|}$, where

$$\bar{\mathbf{v}} = \frac{1}{\sqrt{\nu_n}} \Big(\sum_{b:\mathcal{R}_j^r[b]\ni \mathbf{k}_j^r[n]} \nu_b \, \mathbf{h}_{\mathbf{k}_j^r[b]} \mathbf{h}_{\mathbf{k}_j^r[b]}^H + N_0 \mathbf{I} \Big)^{-1} \mathbf{h}_{\mathbf{k}_j^r[n]} \tag{22}$$

is the optimal RX beamformer solution for the dual uplink channel, for a fixed set of power values ν_n. Plugging (22)

[9]We suspect that altering the placement scheme to remove this unevenness may improve the achievable rate due to the optimization problem's max-min structure. Removing this unevenness would require substantial changes to the scheme, however, and is part of the ongoing research.

into (21), we can write the uplink SINR compactly as

$$\gamma_n = \nu_n \mathbf{h}_{\mathbf{k}_j^r[n]}^H \Big(\sum_{b:\mathcal{R}_j^r[b] \ni \mathbf{k}_j^r[n]} \nu_b \, \mathbf{h}_{\mathbf{k}_j^r[b]} \mathbf{h}_{\mathbf{k}_j^r[b]}^H + N_0 \mathbf{I} \Big)^{-1} \mathbf{h}_{\mathbf{k}_j^r[n]}. \tag{23}$$

Now, for a fixed $\bar{\gamma}$, we can iteratively solve for optimal ν_n values resulting in the minimum total power, using the joint update method in [47]. This involves updating ν_n values via:

$$\begin{aligned} \nu_n^{(m)} &= \frac{\bar{\gamma}}{\gamma_n^{(m-1)}} \nu_n^{(m-1)} \\ &= \frac{\bar{\gamma}}{\mathbf{h}_{\mathbf{k}_j^r[n]}^H \Big(\sum_{b:\mathcal{R}_j^r[b] \ni \mathbf{k}_j^r[n]} \nu_b^{(m-1)} \, \mathbf{h}_{\mathbf{k}_j^r[b]} \mathbf{h}_{\mathbf{k}_j^r[b]}^H + N_0 \mathbf{I} \Big)^{-1} \mathbf{h}_{\mathbf{k}_j^r[n]}}, \end{aligned} \tag{24}$$

until for every $n \in [t+\alpha]$, $\gamma_n = \bar{\gamma}$. Note that the convergence of the joint update method in (24) can be proved by the standard interference function approach [48].

So, in summary, we use an outer iterative loop to find the target SINR value $\bar{\gamma}$ for which the power constraint is met, and an internal iterative loop to find the optimal power coefficients for any given $\bar{\gamma}$. After the outer loop is converged, we calculate $\mathbf{v}_{\mathcal{R}_j^r(n)}$, and then we can find max-min optimal downlink beamformers using $\mathbf{w}_{\mathcal{R}_j^r(n)} = \rho_n \mathbf{v}_{\mathcal{R}_j^r(n)}$, where ρ_n is the downlink power associated with $\mathbf{w}_{\mathcal{R}_j^r(n)}$. To compute ρ_n, we first define $\mathbf{a} = [a_1 \ a_2 \ \ldots \ a_{t+\alpha}]$ and $\mathbf{D}$ to be a diagonal matrix of elements a_1, ..., $a_{t+\alpha}$, where

$$a_n = \frac{\gamma_n}{(1+\gamma_n)|\mathbf{h}_{\mathbf{k}_j^r(n)}^H \mathbf{v}_n|^2}. \tag{25}$$

Then, we define $\mathbf{G}$ as a matrix of elements $g_{n,b}$, $n, b \in [t+\alpha]$, where $g_{n,b} = |\mathbf{h}_{\mathcal{R}_j^r(n)}^H \mathbf{v}_b|^2$ if either $b = n$ or $\mathcal{R}_j^r[b] \ni \mathbf{k}_j^r[n]$, and $g_{n,b} = 0$ otherwise. Finally, defining $\boldsymbol{\rho} = [\rho_1 \ \rho_2 \ \ldots \ \rho_n]$, we have

$$\boldsymbol{\rho} = (\mathbf{I} - \mathbf{DG})^{-1} N_0 \mathbf{a}. \tag{26}$$

F. Further Reduction in Subpacketization

Interestingly, with appropriate modifications, it is possible to further reduce the subpacketization requirement of cyclic caching by a factor of $(gcd(K,t,\alpha))^2$. This not only reduces the subpacketization requirement (and hence, the implementation complexity) considerably but also enables the subpacketization (and complexity) to be adjusted by tuning the α (and K) parameter. For notation simplicity, let us simply use $\phi = gcd(K,t,\alpha)$ to represent the reduction factor. To achieve the reduced subpacketization, we use a user grouping technique inspired by [9]. The idea is to split users into groups of size ϕ and assume each group is equivalent to a *virtual* user. Then, we consider a virtual network consisting of these virtual users, in which the coded caching and spatial multiplexing gains are $\frac{t}{\phi}$ and $\frac{\alpha}{\phi}$, respectively. Finally, we apply the coded caching scheme proposed in Sections III-A and III-B to the virtual network, and *elevate* the resulting cache placement and delivery schemes to be applicable in the original network. The elevation procedure, which is thoroughly explained in Appendix V-B, is designed to achieve the maximum DoF of $t+\alpha$ without any increase in the required subpacketization. As a result, the elevated scheme would require the same subpacketization as the scheme applied to the virtual network, which is $\frac{K}{\phi}\big(\frac{t}{\phi} + \frac{\alpha}{\phi}\big)$, as there exist $\frac{K}{\phi}$ virtual users in the virtual network. Here, we explain the proposed procedure with the help of one example.

Example 6: Assume a network scenario with $K = 8$, $t = 2$ and $\alpha = 4$. In this case, $\phi = 2$ and the resulting virtual network has $K' = \frac{K}{\phi} = 4$ virtual users, coded caching gain of $t' = \frac{t}{\phi} = 1$ and spatial multiplexing gain of $\alpha' = \frac{\alpha}{\phi} = 2$. Assume the virtual users correspond to user groups $v_1 = \{1,2\}$, $v_2 = \{3,4\}$, $v_3 = \{5,6\}$ and $v_4 = \{7,8\}$. Applying the proposed cyclic caching scheme, the cache placement in the virtual network is $\mathcal{Z}(v_{k'}) = \{W_{k'}^q \mid W \in \mathcal{F}, q \in$ [3]$\}$, $\forall k' \in$ [4], and the corresponding subpacketization requirement is $K'(t'+\alpha') = 12$. Data delivery is performed in four rounds, where three transmissions are done during each round. The first transmission vector in the first round is built as

$$\mathbf{x}_1^{1'} = \mathbf{w}'_{v_3} W_2^{1'}(v_1) + \mathbf{w}'_{v_3} W_1^{1'}(v_2) + \mathbf{w}'_{v_2} W_1^{1'}(v_3), \tag{27}$$

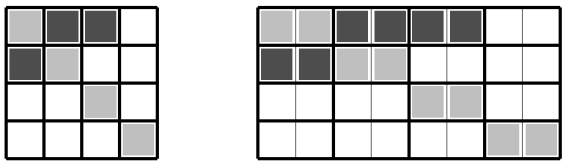
and other transmission vectors are also built similarly. Now, to elevate the cache placement to be applicable to the original network, we simply bind the cache content of each user in the original network with its corresponding virtual user in the virtual network. So, the cache placement for the original network is

$$\begin{aligned} \mathcal{Z}(1) = \mathcal{Z}(2) = \mathcal{Z}(v_1) &= \Big\{ W_1^q \mid W \in \mathcal{F}, q \in \text{ [3]} \Big\}, \\ \mathcal{Z}(3) = \mathcal{Z}(4) = \mathcal{Z}(v_2) &= \Big\{ W_2^q \mid W \in \mathcal{F}, q \in \text{ [3]} \Big\}, \\ \mathcal{Z}(5) = \mathcal{Z}(6) = \mathcal{Z}(v_3) &= \Big\{ W_3^q \mid W \in \mathcal{F}, q \in \text{ [3]} \Big\}, \\ \mathcal{Z}(7) = \mathcal{Z}(8) = \mathcal{Z}(v_4) &= \Big\{ W_4^q \mid W \in \mathcal{F}, q \in \text{ [3]} \Big\}. \end{aligned}$$

Elevating the delivery procedure is more complex. Let us consider the first transmission of the first round, as provided in (27)*. The first term in the transmission vector, i.e., $W_2^{1'}(v_1)$, is suppressed at user v_3, which is equivalent to users $\{5,6\}$ in the original network. However, $W_2^{1'}(v_1)$ itself is combined of two subpackets destined to users 1 and 2, i.e., $W_2^1(1)$ and $W_2^1(2)$. The interference from these two subpackets should also be suppressed at users 2 and 1, which requires the interference indicator set for the first term to be $\{5,6\} \cup \{2\}$ and the one for the second term to be $\{5,6\} \cup \{1\}$. Following the same procedure for the second and third terms, the equivalent transmission vector for the original network will be*

$$\begin{aligned} \mathbf{x}_1^1 = \mathbf{w}_{5,6,2} W_2^1(1) + \mathbf{w}_{5,6,1} W_2^1(2) + \mathbf{w}_{5,6,4} W_1^1(3) \\ + \mathbf{w}_{5,6,3} W_1^1(4) + \mathbf{w}_{3,4,6} W_1^1(5) + \mathbf{w}_{3,4,5} W_1^1(6). \end{aligned} \tag{28}$$

Other transmissions are built similarly. Overall, the algorithm requires subpacketization of 12 and delivers data in 12 transmissions. In comparison, applying cyclic caching without user grouping requires subpacketization of $K(t+L) = 48$, and the number of transmissions would be 48. A graphical



(a) Virtual Network (b) Original Network

Fig. 5. Graphical illustration of transmission vector in Example 6.

representation for the first transmission of the first round, for both virtual and original networks, is provided in Figure 5.

IV. Complexity and Performance Analysis

A. Complexity Analysis

For two main reasons, the subpacketization value is an important complexity indicator for any coded caching scheme. First, it indicates the number of smaller parts each file must be split into for the scheme to work correctly. As argued in [9], the exponentially growing subpacketization of conventional coded caching schemes can make their implementation infeasible, even for networks with a moderate number of users. Second, a scheme with smaller subpacketization *generally* requires a smaller number of transmissions, and consequently, fewer beamformer design problems to be solved. For comparison, cyclic caching requires subpacketization and transmission count of $\frac{K(t+\alpha)}{\phi^2_{K,t,\alpha}}$ and $\frac{K(K-t)}{\phi^2_{K,t,\alpha}}$, respectively. Both these numbers are considerably smaller than the original multi-antenna scheme in [5], which requires subpacketization and transmission count of $\binom{K}{t}\binom{K-t-1}{L-1}$ and $\binom{K}{t+L}$, respectively.

In addition to the performance and complexity benefits of reduced subpacketization, cyclic caching also has the critical advantage of relying on unicasting only, unlike other traditional schemes that rely on high-order multicast messages (e.g., using XOR-ed codewords). Of course, as discussed in [42], removing multicasting causes performance loss. However, it enables high-performance optimized (MMSE-type) beamformers to be applicable with low complexity, even for large networks with a large sum-DoF. In the following subsection, we provide simulation results for large networks with up to $K = 100$ users, in which optimized beamformers are used. To the best of our knowledge, this is the first time a multi-antenna coded caching scheme has been applied with optimized beamformers to such a large network with a large sum-DoF.

From another perspective, cyclic caching also removes the requirement of decoding multiple data parts jointly at the same user during a single transmission. As a result, there is no need for complex receiver schemes such as successive interference cancellation (SIC). Cyclic caching requires that every user in the target group decodes only one message during every transmission, while in [5], $\binom{t+L-1}{t}$ terms must be jointly decoded. Of course, this complexity reduction is also possible by splitting each transmission with overlapping multicast messages into multiple TDMA intervals, as shown in [26]. However, it comes with a substantial subpacketization increase.

Interestingly, cyclic caching also enables tuning α and K parameters jointly to reduce both subpacketization and transmission count. This reduction is useful especially for large networks with a large number of users K, for which the complexity of coded caching schemes is critically limiting their practical implementation [9]. Selecting α to be smaller than the antenna count is straightforward and explained in [26]. However, in order to tune K, we consider a set of K_f additional *phantom* users and tune both α and K_f to maximize $gcd(K+K_f, t, \alpha)$. K_f phantom users are then omitted during all the subsequent transmissions. Of course, tuning either parameter comes with a DoF loss. For α, this is not necessarily an issue, especially when the communication is at finite-SNR. In [26], it is shown that by choosing $\alpha < L$, one can obtain an improved beamforming gain, which considerably improves the performance at the finite-SNR regime. The joint impact of α and K_f tuning is studied through numerical simulations in Section IV-B.

In principle, the reduced subpacketization scheme of Section III-F can be applied by splitting users into groups of size $Q > 1$, such that $gcd(K, t, \alpha)$ is divisible by Q. This enables selecting several subpacketization levels, between $K(t+\alpha)$ and $\frac{K(t+\alpha)}{gcd(K,t,\alpha)^2}$. However, as we show later, the performance of cyclic caching is almost intact regardless of the selected subpacketization, and hence, it makes sense to select the largest possible $gcd(K, t, \alpha)$ value.

In Table II, we have summarized the key advantages of the cyclic caching scheme. Moreover, in Table III, we have compared the complexity order, in terms of both subpacketization requirement and the total number of transmissions, for the multi-antenna scheme of [5] (M-S), cyclic caching without user grouping (LIN), cyclic caching with user grouping (RED), the original group-based scheme in [9] (L-E), and the recently proposed scheme in [40] (M-B). In Table III, the complexity order is provided for two cases where the global cache ratio $t = \frac{KM}{N}$ is fixed and when it scales with the number of users K. It t does not scale with K, we have simply used

$$\binom{K}{t} = \frac{K!}{t!(K-t)!} = O(K^t). \tag{29}$$

However, if t scales with K (i.e., if $\gamma = \frac{M}{N}$ does not scale with K), this order approximation is no longer valid. In this case, to approximate $\binom{K}{t}$, we can consider another problem where we repeat a binary experiment with the success probability of γ for K times. As K grows large, according to the law of large numbers, the number of *typical* sequences (i.e., sequences with $K\gamma = t$ success outcomes) would be $\binom{K}{t}$. However, from information theory, we also know that the number of typical sequences approaches $H(\gamma)$, where

$$H(\gamma) = -\gamma \log_2 \gamma - (1-\gamma)\log_2(1-\gamma)$$

is the entropy function. Hence, when t scales with K we can use the approximation $\binom{K}{t} = O(2^{KH(\gamma)})$. From Table III, it is clear that the complexity order of the proposed cyclic caching scheme is considerably smaller than all other schemes (linear if t does not scale with K, and quadratic otherwise).

Finally, in Table IV, we have respectively compared the subpacketization requirement and the transmission count in some different network setups for M-S, LIN, RED, L-E, and M-B schemes. In the table, many entries are left empty for

TABLE II

ADVANTAGES OF THE CYCLIC CACHING SCHEME

Reduced subpacketization	Cyclic caching enables achieving the sum-DoF of $t+\alpha$, with the reduced subpacketization of $\frac{K(t+\alpha)}{\phi^2_{K,t,\alpha}}$.
Reduced number of transmissions	With cyclic caching, delivery to all users is completed with only $\frac{K(K-t)}{\phi^2_{K,t,\alpha}}$ transmissions.
Relying only on unicasting	Cyclic caching relies on unicasting only. As a result, optimized MMSE-type beamformers can be implemented with much lower complexity using the uplink-downlink duality. This improves the performance at low- and mid-SNR, compared with ZF beamforming.
No MAC decoding	Cyclic caching removes the requirement of MAC decoding, thus eliminating the necessity of complex receiver structures such as successive interference cancellation (SIC).
Controlling the complexity	The subpacketization of cyclic caching can be controlled by tuning the $\phi_{K,t,\alpha}$ parameter, which is possible by adjusting α and also by considering a set of K_f phantom users.

TABLE III

COMPLEXITY ORDER OF DIFFERENT SCHEMES WHEN t IS FIXED AND WHEN IT SCALES WITH K, $\gamma = \frac{M}{N}$, $H(\cdot)$ IS THE ENTROPY FUNCTION

Scheme	M-S		LIN		RED		L-E		M-B	
Scaling of t	fixed	scaling	fixed	scaling	fixed	scaling	fixed	scaling	fixed	scaling
Subpacketization	$O(K^t K^{L-1})$	$O(2^{KH(\gamma)}K^{L-1})$	$O(K)$	$O(K^2)$	$O(K)$	$O(K^2)$	$O(K^{t/L})$	$O(2^{KH(\gamma)/L^2})$	$O(K^t)$	$O(2^{KH(\gamma)})$
Trans. Count	$O(K^{t+L})$	$O(2^{KH(\gamma)})$	$O(K^2)$	$O(K^2)$	$O(K^2)$	$O(K^2)$	$O(K^{t/L+1})$	$O(2^{KH(\gamma)/L})$	$O(K^{t+1})$	$O(2^{KH(\gamma)})$

TABLE IV

COMPLEXITY COMPARISON FOR SOME EXAMPLE NETWORK SETUPS

					Required Subpacketization					Required number of transmissions (I)				
K	t	L	α	K_f	M-S	LIN	RED	L-E	M-B	M-S	LIN	RED	L-E	M-B
8	2	5	2	0	140	32	8	4	-	70	48	12	6	-
8	2	5	4	0	280	48	12	-	28	28	48	12	-	28
8	2	5	5	0	140	56	56	-	-	8	48	48	-	-
30	4	8	4	0	$> 10^7$	240	60	-	-	$> 10^6$	780	195	-	-
30	4	8	4	2	$> 10^8$	256	16	8	-	$> 10^7$	832	52	28	-
30	4	8	6	0	$> 10^9$	300	75	-	$> 10^4$	$> 10^7$	780	195	-	$> 10^4$
100	15	30	15	0	$> 10^{32}$	3000	120	-	-	$> 10^{32}$	8500	340	-	-
100	15	30	15	5	$> 10^{33}$	3150	14	7	-	$> 10^{26}$	9450	42	21	-
100	15	30	17	0	$> 10^{34}$	3200	3200	-	$> 10^{17}$	$> 10^{26}$	8500	8500	-	$> 10^{17}$
400	50	200	100	0	$> 10^{153}$	$> 10^4$	24	-	-	$> 10^{113}$	$> 10^5$	56	-	-

L-E and M-B schemes due to their tight restrictions on the network parameters. The L-E scheme is originally designed for networks in which $t \geq \alpha$ and α divides both t and K, while M-B requires $\frac{t+\alpha}{t+1}$ to be an integer.

From Table IV, it is clear that except for the L-E scheme, the RED scheme has the lowest subpacketization requirement among all the schemes. Also, regarding the number of transmissions, except for a specific case (the third row in the table), LIN and RED outperform all other schemes. It can be seen that, even when the user count is as low as $K = 30$, the M-S scheme becomes infeasible due to the substantial values for both the subpacketization and the transmission count. On the other hand, it is possible to implement RED even for a very large network of $K = 400$ users in which the spatial DoF is $\alpha = 100$. The results also show how proper tuning of α and K_f can help further reduce both the complexity and subpacketization in the RED scheme.

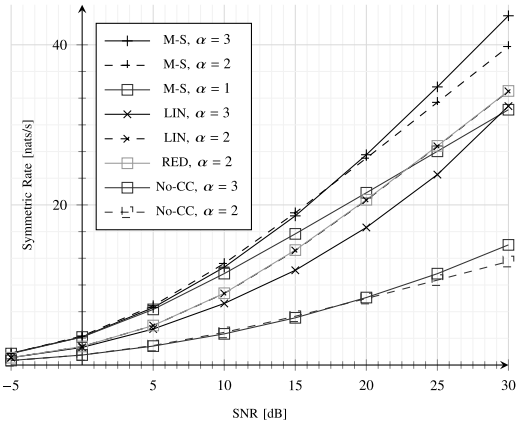
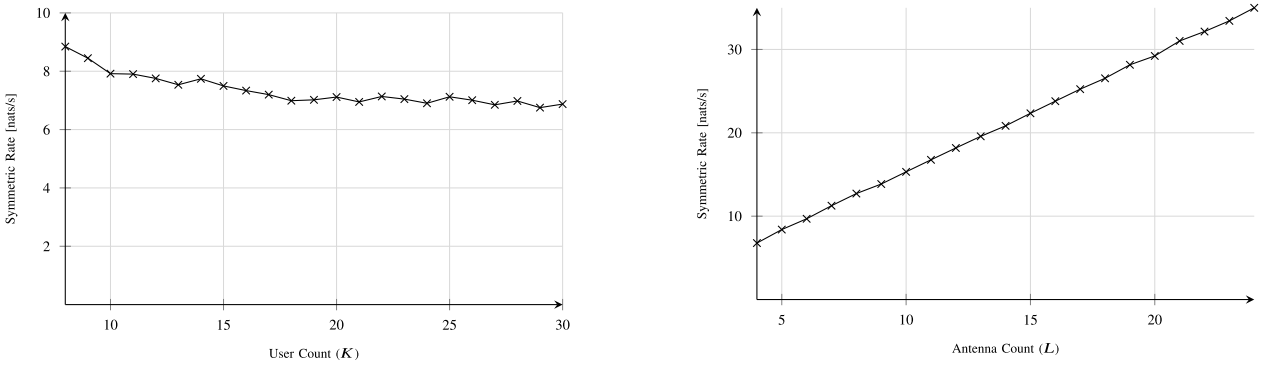


Fig. 6. M-S vs. LIN vs. RED vs. No-CC rate; $K = 6$, $t = 2$, $L = 3$.

B. Simulation Results

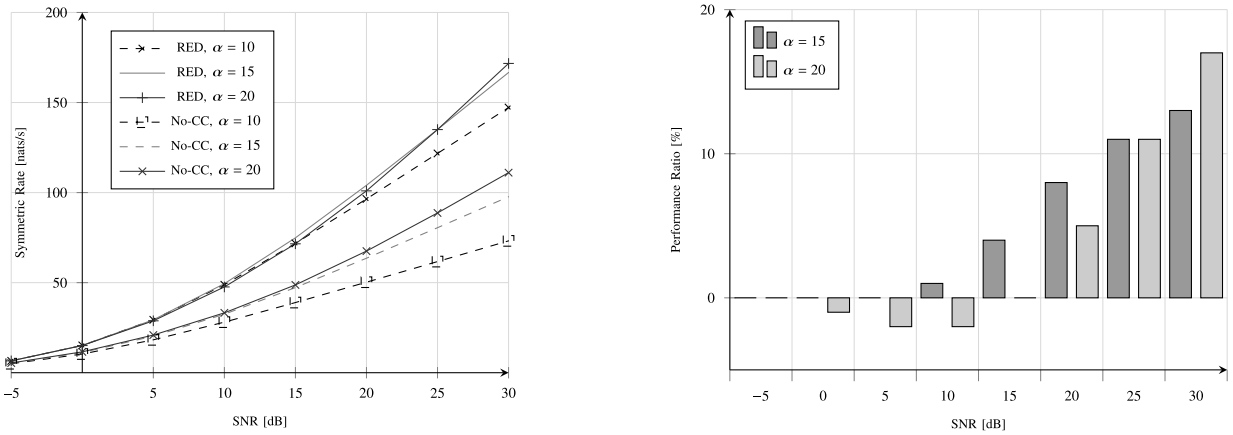
We use numerical simulations to compare the performance of cyclic caching with other schemes. The symmetric rate, as defined in (1), is used as the comparison metric. The L-E and M-B schemes are ignored in the simulations as they require tight restrictions on network parameters to work without significant performance (DoF) loss. Moreover, for the sake of comparison, we also consider a baseline scheme without coded caching, denoted by No-CC, in which only the local caching gain at each user is attained together with the spatial multiplexing gain. In the baseline scheme, we create K transmission vectors, where users $\{1, 2, \ldots, \alpha\}$ are served during the first transmission ($i = 1$), while for $i > 1$,



(a) Performance vs K, $L = 4$

(b) Performance vs L, $K = 30$

Fig. 7. Performance of RED vs. K and L parameters, $t = 2$, $\alpha = L$, SNR$=$ 10dB.



(a) Performance vs α

(b) Performance ratio w.r.t. $\alpha = 10$

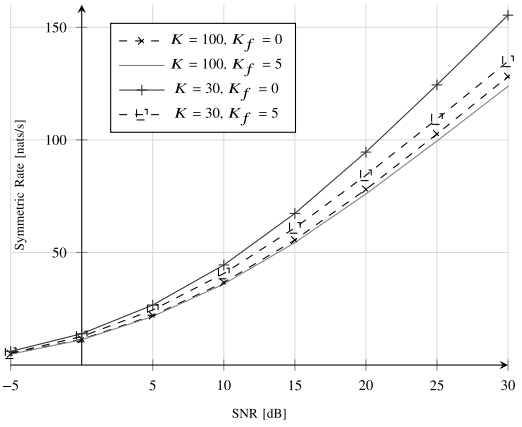
Fig. 8. Performance of RED for a large network, $K = 100$, $t = 10$, $L = 25$.

the served user indices are a circular shift of the user indices targeted at transmission $i - 1$. For all the simulations, we use max-min-SINR optimal (MMSE-type) beamformers, found through solving (19). For LIN, RED, and No-CC schemes, the beamformers are designed with the uplink-downlink duality solution described in Section III-E. For the M-S scheme, we use the more complex SCA method detailed in [26].

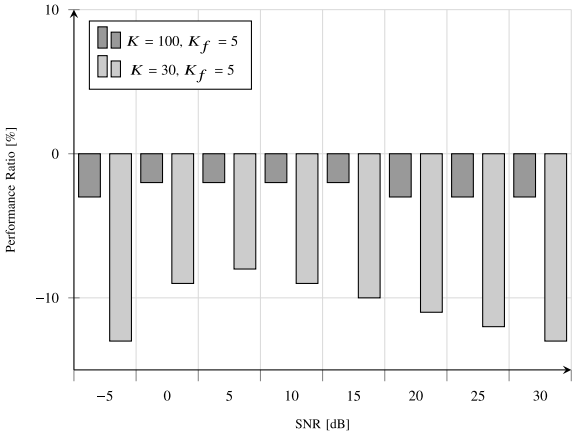
As discussed earlier, the complexity of the M-S scheme makes its implementation infeasible even for moderate-sized networks. In order to be able to compare all the schemes, we consider a small network of $K = 6$ users. The performance comparison results are provided in Figure 6. It can be seen that the performance values of LIN and RED schemes lie between M-S and No-CC. M-S provides better performance because it benefits from a *multicasting* gain, i.e., a single codeword (created with the XOR operation) benefits all the users in a multicast group. This multicasting effect is explained in more detail in [42], with the help of the so-called efficiency index parameter. On the other hand, the No-CC scheme has the worst performance as it lacks the coded caching gain entirely. It should also be noted that choosing a smaller α value improves the performance of both M-S and LIN schemes at the lower SNR range, while this effect is more prominent for the LIN scheme.[10] This result is in line with the findings of [26], [41], where a smaller α is shown to improve the performance at the finite-SNR regime due to a better beamforming gain. Finally, Figure 6 demonstrates that the RED scheme provides the same performance as the LIN scheme but with lower complexity. This near-identical performance is a result of the two schemes having a very similar coding and interference-cancellation structure.

Unfortunately, although the M-S scheme has superior performance, its high complexity makes the implementation infeasible even for slightly larger networks. For example, for the small network of 6 users considered in Figure 6, with our simulation setup, the required time for simulating the M-S scheme is $O(10^3)$ times larger than the LIN scheme. As the network size grows, this ratio between the simulation times also grows exponentially. On the other hand, the simulation time for the RED scheme is roughly four times smaller than LIN. This is in line with the fact that for the considered

[10]Note that when $\alpha = 1$, placement and delivery algorithms in M-S scheme are the same as the original scheme of [2]. The beamformer design also concedes with the max-min fair beamforming in [5].



(a) Performance vs K_f



(b) Performance ratio w.r.t. $K_f = 0$

Fig. 9. The effect of the K_f parameter, RED scheme, $t = 7$, $L = 20$, $\alpha = 14$.

network with $K = 6, t = 2, \alpha = 2$, subpacketization and transmission count are reduced by a factor of $gcd(K, t, \alpha)^2 = 4$.

In Figure 7, we have analyzed the performance of the RED scheme with respect to K and L parameters while assuming $t = 2$ and $\alpha = L$. From Figure 7a, we see that with the DoF value $t + L$ fixed, the performance slightly reduces as K increases. This is because we have assumed a fixed $t = \frac{KM}{N}$ value, which means the local caching gain $\frac{M}{N}$ is reduced as K is increased. On the other hand, from Figure 7b, we see that increasing L results in a linear increase in rate. This is simply because increasing L improves DoF, and accordingly, the system performance.

Figure 8 illustrates the performance of the RED scheme for a large network with $K = 100$,$t = 10$, and $L = 25$. To the best of our knowledge, this is the first time a coded caching scheme is applied with optimized beamformers for such a large network. In Figure 8a, the results for the No-CC scheme are included for comparison. It can be verified that decreasing α from 20 to 10 gives a small performance boost at the low- to moderate-SNR regime (< 15dB) due to improved beamforming gain. On the other hand, at larger SNR values, the $\alpha = 20$ setup performs much better due to the increased spatial multiplexing gain. These are in line with the findings in [26].

Finally, the complexity reduction effect of the K_f parameter is analyzed in Figure 9 for two network scenarios of size $K_1 = 100$ and $K_2 = 30$ users. In both networks, we have assumed $t = 7$, $L = 20$ and $\alpha = 14$. Without any phantom users, $gcd(K, t, \alpha) = 1$ and the subpacketization values for the two networks are $S_1 = 2100$ and $S_2 = 630$, respectively. However, by adding only $K_f = 5$ phantom users, the common denominator becomes $gcd(K + K_f, t, \alpha) = 7$, and hence, the subpacketization values are reduced to $S_1 = 45$ and $S_2 = 15$, respectively. Interestingly, the decrease in the achievable rate due to adding K_f phantom users is relatively minor for both networks. In fact, for the larger network, the performance loss is less than 4% over the entire SNR range, while for the smaller network, the deterioration is less than 15%. On the other hand, adding the extra phantom users reduces the simulation time in our setup by a factor of ~ 10.

V. Conclusion and Future Work

We proposed cyclic caching, a novel low-complexity, high-performance coded caching scheme for large-scale cache-aided MISO networks where the spatial DoF is not smaller than the coded caching gain. For a fixed caching gain, cyclic caching achieves theoretical sum-DoF optimality with a subpacketization requirement that scales linearly with the number of users in the network. Moreover, it delivers all the requested files with a relatively small number of transmissions and does not require the users to decode multiple messages simultaneously. As a result, the scheme can be employed in large networks with many users, at least one order of magnitude larger than other well-known multi-antenna coded caching schemes.

Our proposed scheme is centralized, i.e., the server knows the number and identity of the users beforehand and dictates the cached content for each user. A decentralized extension for this work should handle scenarios where such information is unavailable. However, the classic decentralized approach (cf. [13]) where users cache at random a set of bits from the library heavily degrades the performance in the practical finite file size regime considered here. Moreover, our RED scheme depends largely on the existence of similar cached contents at different users, a property that very unlikely is preserved in a fully decentralized approach. Hence, we believe that developing a semi-decentralized scheme, similar to [16] where each user picks at random one of the pre-defined cache states, can result in a good performance even in the finite file size regime. Other topics to be considered include extending the applicability of the scheme to the $t > L$ regime and investigating the effect of the non-equal number of interference terms for different users.

Appendix

A. More Detailed Analysis of the Delivery Phase

For the delivery and decoding procedures presented in Sections III-B and III-C, we here show that at the end of the K rounds, each user successfully decodes all its missing data parts. We recall that for the placement matrix $\mathbf{V}$, if $\mathbf{V}[p, k] = 0$

for some $p \in [K]$ and $k \in [K]$, then the packet $W_p(k)$, which is comprised of $t+\alpha$ smaller subpackets of the form $W_p^q(k)$, must be delivered to user k. By construction of the user index and packet index vectors $\mathbf{k}_j^r$ and $\mathbf{p}_j^r$, where $r \in [K]$ and $j \in [t+\alpha]$, it is easy to see that for any $p \in [K]$ and $k \in [K]$, if $\mathbf{V}[p,k] = 0$ there exists always a transmission vector that contains one of the $t+\alpha$ subpackets represented by $\mathbf{V}[p,k]$; i.e. there exists always a triple (j,r,n) such that $p = \mathbf{p}_j^r[n]$ and $k = \mathbf{k}_j^r[n]$. Therefore, it is sufficient to show that if $\mathbf{V}[p,k] = 0$ for some pair (p,k), there exist exactly $t+\alpha$ triples (j,r,n) for which $p = \mathbf{p}_j^r[n]$ and $k = \mathbf{k}_j^r[n]$.

Without loss of generality, let us consider the first round of the delivery scheme. The vector $\mathbf{k}_j^1$ contains the index of the users targeted at transmission j of this round. Let us split the user indices in $\mathbf{k}_j^1$ into the following two vectors

$$\mathbf{c}_j^1 = [1:t], \quad \mathbf{s}_j^1 = [(([1:\alpha]+j-1)\%(K-t))+t].$$

Similarly, we split the packet indices in $\mathbf{p}_j^1$ into the following two vectors

$$\mathbf{r}_j^1 = [((t+j-[1:t])\%(K-t))+[1:t]], \quad \mathbf{d}_j^1 = \mathbf{e}(\alpha).$$

Now let us consider the set of points in the tabular representation corresponding to the vector pair $(\mathbf{r}_1^j, \mathbf{c}_1^j)$; i.e., the set of points in which the row index is taken from $\mathbf{r}_1^j$ and column index is taken from $\mathbf{c}_1^j$. We will use $\{(\mathbf{r}_1^j, \mathbf{c}_1^j)\}$ to denote this set. From the graphical example in Section III-D, we recall that $\{(\mathbf{r}_1^j, \mathbf{c}_1^j)\}$ is an element-wise circular shift of $\{(\mathbf{r}_1^1, \mathbf{c}_1^1)\}$, over the non-shaded cells of the tabular representation and in the vertical direction. Similarly, $\{(\mathbf{d}_1^j, \mathbf{s}_1^j)\}$ is an element-wise circular shift of $\{(\mathbf{d}_1^1, \mathbf{s}_1^1)\}$, over the non-shaded cells and in the horizontal direction. As a result, in round 1, the following two statements hold:

- for every user $k \in [K]$ and packet index $p \in [K]$, if k is in $\mathbf{c}_1^1 = \mathbf{c}_2^1 = \ldots = \mathbf{c}_{K-t}^1$ and p is in $[\mathbf{r}_1^1; \ldots; \mathbf{r}_{K-t}^1]$, there exists exactly one transmission index $j \in [K-t]$, such that the j-th transmission of the first round delivers $W_p^q(k)$ to user k, for some subpacket index $q \in [t+\alpha]$. In other words there exists exactly one triple $(j,1,n)$ for the pair (p,k);
- for every user $k \in [K]$, if k is in $[\mathbf{s}_1^1; \mathbf{s}_2^1; \ldots; \mathbf{s}_{K-t}^1]$, there exist exactly α transmissions in the first round that deliver $W_1^q(k)$ to user k, each with a distinct subpacket index $q \in [t+\alpha]$. In other words, there exist exactly α triples $(j,1,n)$, for the pair $(1,k)$.

These statements also hold for other transmission rounds, i.e., transmission round r where $1 < r \leq K$. However, for any $k,p \in [K]$ such that $\mathbf{V}[p,k] = 0$, there exists exactly one round r for which k is in $[\mathbf{s}_1^r; \mathbf{s}_2^r; \ldots; \mathbf{s}_{K-t}^r]$, while there exist t different rounds r for which k is in $\mathbf{c}_1^r = \mathbf{c}_2^r = \ldots = \mathbf{c}_{K-t}^r$. Hence, for any $k,p \in [K]$ such that $\mathbf{V}[p,k] = 0$, there exist

$$\alpha \times 1 + 1 \times t = \alpha + t$$

triples (j,r,n) for the pair (p,k). This clarifies the correctness of the delivery algorithm proposed in Section III-B.

B. Reducing Subpacketization by a Factor of $\phi^2_{K,t,\alpha}$

To reduce the subpacketization requirement, we apply a user grouping mechanism, inspired by [9]. For notation simplicity, let us use simply use ϕ to represent $\phi_{K,t,\alpha}$. The idea is to split users into groups of size $\phi_{K,t,\alpha}$ and assume each group is equivalent to a *virtual* user. Then, we consider a virtual network consisting of these virtual users, in which the coded caching and spatial multiplexing gains are $\frac{t}{\phi}$ and $\frac{\alpha}{\phi}$, respectively. Finally, we apply the coded caching scheme proposed in Sections III-A and III-B to the virtual network, and *elevate* the resulting cache placement and delivery schemes to be applicable in the original network. Here, we provide a detailed description of the elevation procedure.

1) Cache Placement: Assume the original network is given with K users, coded caching gain of t, and spatial DoF of α. We first split the set of users $[K]$ into $K' = \frac{K}{\phi}$ disjoints groups $v_{k'}$, $k' \in [K']$, where all groups have the same number of ϕ users. Without loss of generality, we assume each group $v_{k'}$ corresponds to the set of users

$$v_{k'} \triangleq [\phi * (k'-1) + 1 : \phi * k'], \quad \forall k' \in [K']. \tag{30}$$

Next, we assume each user group is equivalent to a virtual user with cache size of Mf bits, and the virtual users form a new virtual network in which the spatial DoF is $\alpha' = \frac{\alpha}{\phi}$. For this virtual network, we use the same cache placement algorithm presented in Section III-A, and set the cache content of every user in group $v_{k'}$ to be the same as the cache content of the virtual user corresponding to $v_{k'}$. The total subpacketization is then $K'(t'+\alpha')$, where $t' = \frac{t}{\phi}$.

2) Delivery Phase: During the delivery phase, we first create transmission vectors for the virtual network, and then *elevate* them to be applicable in the original network. Following (2), the transmission vector i for the virtual network is built as $\mathbf{x}_i' = \sum_{v_{k'} \in \mathcal{X}_i'} \mathbf{w}_i'(v_{k'}) X_i'(v_{k'})$, in which $\mathcal{X}_i'$ is the set of virtual users targeted at transmission i, $X_i'(v_{k'})$ denotes the data part targeted to the virtual user $v_{k'}$ during the same transmission, and $\mathbf{w}_i'(v_{k'})$ is the beamformer vector assigned to $X_i'(v_{k'})$. In order to elevate $\mathbf{x}_i'$ for the original network, we first notice than each virtual user represents a set of ϕ original users, and hence, using (30), the set of targeted users during transmission i for the original network is

$$\mathcal{X}_i = \bigcup_{v_{k'} \in \mathcal{X}_i'} [\phi * (k'-1) + 1 : \phi * k']. \tag{31}$$

Following the discussions in Section III-E, every beamformer vector $\mathbf{w}_i'(v_{k'})$ is built to suppress unwanted terms at $\alpha' - 1$ virtual users. Let us denote the set of such virtual users as $\mathcal{R}_i'(v_{k'})$, where $|\mathcal{R}_i'(v_{k'})| = \alpha' - 1$. The goal is then to find the respective set for the original network, denoted by $\mathcal{R}_i(k)$, for $k \in [K]$. As the spatial DoF for the original network is α, it is possible to suppress undesired terms at $\alpha - 1$ original users; i.e. $|\mathcal{R}_i(k)| = \alpha - 1$. Without loss of generality, let us assume that user k is in the respective group of $v_{k'}$ (every user in the original network has one counterpart in the virtual network). In the original network, for the interference to be suppressed, the following conditions should be met:

1) For every $v_{\hat{k}'} \in \mathcal{R}'_i(v_{k'})$, $\mathcal{R}_i(k)$ should include all the users in the respective group of $v_{\hat{k}'}$; i.e. $\mathcal{R}_i(k)$ should include all the users in $[\phi * (\hat{k}' - 1) + 1 : \phi * \hat{k}']$;
2) $\mathcal{R}_i(k)$ should include all other users in the respective group of $v_{k'}$; i.e. $\mathcal{R}_i(k)$ should include all the users in $[\phi * (k' - 1) + 1 : \phi * k']\backslash\{k\}$.

Using a formal representation, we have

$$\mathcal{R}_i(k) = [\phi * (k' - 1) + 1 : \phi * k']\backslash\{k\} \bigcup_{v_{\hat{k}'} \in \mathcal{R}'_i(v_{k'})} [\phi * (\hat{k}' - 1) + 1 : \phi * \hat{k}']. \quad (32)$$

In other words, the data part $X_i(k)$ intended for user k at transmission i has to be suppressed not only at $\alpha' - 1$ virtual users (where each virtual user represents ϕ original users), but also at $\phi - 1$ original users in the equivalent group of $v_{k'}$. Hence, the total number of users for which $X_i(k)$ should be suppressed is $(\alpha' - 1)\phi + \phi - 1$. Substituting $\alpha' = \frac{\alpha}{\phi}$, we have

$$|\mathcal{R}_i(k)| = (\frac{\alpha}{\phi} - 1)\phi + \phi - 1 = \alpha - 1. \quad (33)$$

Now, we can elevate $\mathbf{x}'_i$ to be applicable in the original network. All we need to do is to use (31) to substitute the target user set $\mathcal{X}'_i$ with $\mathcal{X}_i$, and replace $\mathbf{w}'_i(v_{k'})X'_i(v_{k'})$ with

$$\sum_{k \in [\phi * (k' - 1) + 1 : \phi * k']} \mathbf{w}_i(k) X_i(k), \quad (34)$$

where $\mathbf{w}_i(k)$ is the beamformer vector designed to suppress unwanted data terms at the user set $\mathcal{R}_i(k)$ as defined in (32), and $X_i(k)$ is the data part intended for user k at transmission i. The subpacketization for the virtual network, $K'(t' + \alpha')$, would then be still valid for the original network, indicating a reduction of ϕ^2 compared with the case when no grouping is applied.

REFERENCES

[1] *Cisco Visual Networking Index: Forecast and Trends, 2017–2022*, VNI Cisco, San Jose, CA, USA, 2018.

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

[3] N. Rajatheva *et al.*, "White paper on broadband connectivity in 6G," 2020, *arXiv:2004.14247*. [Online]. Available: http://arxiv.org/abs/2004.14247

[4] 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.

[5] S. P. Shariatpanahi, G. Caire, and B. H. Khalaj, "Physical-layer schemes for wireless coded caching," *IEEE Trans. Inf. Theory*, vol. 65, no. 5, pp. 2792–2807, May 2019.

[6] E. Lampiris, A. Bazco-Nogueras, and P. Elia, "Resolving the feedback bottleneck of multi-antenna coded caching," 2018, *arXiv:1811.03935*. [Online]. Available: http://arxiv.org/abs/1811.03935

[7] K. Shanmugam, M. Ji, A. M. Tulino, J. Llorca, and A. G. Dimakis, "Finite-length analysis of caching-aided coded multicasting," *IEEE Trans. Inf. Theory*, vol. 62, no. 10, pp. 5524–5537, Oct. 2016.

[8] 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.

[9] 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.

[10] Q. Yu, M. A. Maddah-Ali, and A. S. Avestimehr, "The exact rate-memory tradeoff for caching with uncoded prefetching," *IEEE Trans. Inf. Theory*, vol. 64, no. 2, pp. 1613–1617, Feb. 2018.

[11] K. Wan, D. Tuninetti, and P. Piantanida, "On the optimality of uncoded cache placement," in *Proc. IEEE Inf. Theory Workshop*, Jun. 2016, pp. 161–165.

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

[13] M. A. Maddah-Ali and U. Niesen, "Decentralized coded caching attains order-optimal memory-rate tradeoff," *IEEE/ACM Trans. Netw.*, vol. 23, no. 4, pp. 1029–1040, Aug. 2014.

[14] A. M. Girgis, O. Ercetin, M. Nafie, and T. ElBatt, "Decentralized coded caching in wireless networks: Trade-off between storage and latency," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jun. 2017, pp. 2443–2447.

[15] F. Xu and M. Tao, "Fundamental limits of decentralized caching in fog-RANs with wireless fronthaul," 2018, *arXiv:1805.03613*. [Online]. Available: http://arxiv.org/abs/1805.03613

[16] 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.

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

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

[19] J. S. P. Roig, F. Tosato, and D. Gündüz, "Storage-latency trade-off in cache-aided fog radio access networks," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Dec. 2018, pp. 1–6.

[20] N. Naderializadeh, M. A. Maddah-Ali, and A. S. Avestimehr, "Fundamental limits of cache-aided interference management," *IEEE Trans. Inf. Theory*, vol. 63, no. 5, pp. 3092–3107, May 2017.

[21] N. Naderializadeh, M. A. Maddah-Ali, and A. S. Avestimehr, "Cache-aided interference management in wireless cellular networks," *IEEE Trans. Commun.*, vol. 67, no. 5, pp. 3376–3387, May 2019.

[22] E. Piovano, H. Joudeh, and B. Clerckx, "Centralized and decentralized cache-aided interference management in heterogeneous parallel channels," *IEEE Trans. Commun.*, vol. 68, no. 3, pp. 1881–1896, Mar. 2020.

[23] J. Hachem, U. Niesen, and S. N. Diggavi, "Degrees of freedom of cache-aided wireless interference networks," *IEEE Trans. Inf. Theory*, vol. 64, no. 7, pp. 5359–5380, Jul. 2018.

[24] 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.

[25] A. Tölli, S. P. Shariatpanahi, J. Kaleva, and B. Khalaj, "Multicast beamformer design for coded caching," in *Proc. IEEE Int. Symp. Inf. Theory*, Jun. 2018, pp. 1914–1918.

[26] A. Tolli, 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.

[27] E. Lampiris and P. Elia, "Bridging two extremes: Multi-antenna coded caching with reduced subpacketization and CSIT," in *Proc. Workshop Signal Process. Adv. Wireless Commun.*, Jul 2019, pp. 1–5.

[28] J. Zhao, M. M. Amiri, and D. Gündüz, "Multi-antenna coded content delivery with caching: A low-complexity solution," 2020, *arXiv:2001.01255*. [Online]. Available: http://arxiv.org/abs/2001.01255

[29] S. P. Shariatpanahi and B. H. Khalaj, "On multi-server coded caching in the low memory regime," 2018, *arXiv:1803.07655*. [Online]. Available: http://arxiv.org/abs/1803.07655

[30] E. Lampiris, J. Zhang, O. Simeone, and P. Elia, "Fundamental limits of wireless caching under uneven-capacity channels," 2019, *arXiv:1908.04036*. [Online]. Available: http://arxiv.org/abs/1908.04036

[31] I. Bergel and S. Mohajer, "Cache-aided communications with multiple antennas at finite SNR," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 8, pp. 1682–1691, Aug. 2018.

[32] Q. Yan, X. Tang, Q. Chen, and M. Cheng, "Placement delivery array design through strong edge coloring of bipartite graphs," *IEEE Commun. Lett.*, vol. 22, no. 2, pp. 236–239, Feb. 2018.

[33] 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.

[34] 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.

[35] H. H. S. Chittoor, P. Krishnan, and K. V S. Sree, "Subexponential and linear subpacketization coded caching via line graphs and projective geometry," 2020, *arXiv:2001.00399*. [Online]. Available: http://arxiv.org/abs/2001.00399
[36] 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.
[37] J. Michel and Q. Wang, "Placement delivery arrays from combinations of strong edge colorings," *IEEE Trans. Commun.*, vol. 68, no. 10, pp. 5953–5964, Oct. 2020.
[38] A. A. Mahesh and B. Sundar Rajan, "A coded caching scheme with linear sub-packetization and its application to multi-access coded caching," 2020, *arXiv:2009.10923*. [Online]. Available: http://arxiv.org/abs/2009.10923
[39] E. Parrinello, A. Ünsal, and P. Elia, "Fundamental limits of coded caching with multiple antennas, shared caches and uncoded prefetching," *IEEE Trans. Inf. Theory*, vol. 66, no. 4, pp. 2252–2268, Apr. 2019.
[40] S. Mohajer and I. Bergel, "MISO cache-aided communication with reduced subpacketization," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Jun. 2020, pp. 1–6.
[41] E. Lampiris, P. Elia, and G. Caire, "Bridging the gap between multiplexing and diversity in finite SNR multiple antenna coded caching," in *Proc. 53rd Asilomar Conf. Signals, Syst., Comput.*, Nov. 2019, pp. 1272–1277.
[42] M. Salehi, A. Tolli, S. P. Shariatpanahi, and J. Kaleva, "Subpacketization-rate trade-off in multi-antenna coded caching," in *Proc. IEEE Global Commun. Conf.*, Dec. 2019, pp. 1–6.
[43] M. J. Salehi, A. Tolli, and S. P. Shariatpanahi, "Subpacketization-beamformer interaction in multi-antenna coded caching," in *Proc. 2nd 6G Wireless Summit, Gain Edge 6G Era*, 2020, pp. 1–5.
[44] S. A. Vorobyov, A. B. Gershman, and Z.-Q. Luo, "Robust adaptive beamforming using worst-case performance optimization: A solution to the signal mismatch problem," *IEEE Trans. Signal Process.*, vol. 51, no. 2, pp. 313–324, Feb. 2003.
[45] P. Komulainen, "Coordinated multi-antenna techniques for cellular networks: Pilot signaling and decentralized optimization in TDD mode," Ph.D. dissertation, Dept. Commun. Eng., Center Wireless Commun., Univ. Oulu, Oulu, Finland, 2013. [Online]. Available: http://urn.fi/urn:isbn:9789526202815
[46] M. Bengtsson and B. Ottersten, "Optimum and suboptimum transmit beamforming," in *Handbook of Antennas in Wireless Communications*. Boca Raton, FL, USA: CRC Press, 2018, pp. 1–18.
[47] A. Wiesel, Y. C. Eldar, and S. Shamai (Shitz), "Linear precoding via conic optimization for fixed MIMO receivers," *IEEE Trans. Signal Process.*, vol. 54, no. 1, pp. 161–176, Dec. 2005.
[48] R. D. Yates, "A framework for uplink power control in cellular radio systems," *IEEE J. Sel. Areas Commun.*, vol. 13, no. 7, pp. 1341–1347, Sep. 1995.

MohammadJavad Salehi (Member, IEEE) received the B.Sc., M.Sc., and Ph.D. degrees in electrical engineering from the Sharif University of Technology, Tehran, Iran, in 2010, 2012, and 2018, respectively. Since 2019, he has been a Post-Doctoral Researcher with the Center for Wireless Communications (CWC), University of Oulu, Finland. Before starting his career in Oulu, he has also had several years of professional experience in the ICT industry. His current research interests include coded caching and multi-antenna communications.

Emanuele Parrinello (Member, IEEE) received the B.Sc. degree in telecommunication engineering and the M.Sc. degree (Hons.) in communications and computer networks engineering from the Politecnico di Torino in 2015 and 2018, respectively, the M.Sc. degree in mobile communications from the EURECOM, Télécom Paris, in 2018, and the Ph.D. degree from Sorbonne University, France, in 2021. He is currently employed as a Signal Processing Engineer at CEVA, France. His research interests lie in caching networks, network information theory, wireless communication, and signal processing.

Seyed Pooya Shariatpanahi received the B.Sc., M.Sc., and Ph.D. degrees from the Department of Electrical Engineering, Sharif University of Technology, Tehran, Iran, in 2006, 2008, and 2013, respectively. He is currently an Assistant Professor with the School of Electrical and Computer Engineering, College of Engineering, University of Tehran. Before joining the University of Tehran, he was a Researcher with the Institute for Research in Fundamental Sciences (IPM), Tehran. His research interests include information theory, network science, wireless communications, and complex systems. He was a recipient of the Gold Medal at the National Physics Olympiad in 2001.

Petros Elia received the B.Sc. degree from the Illinois Institute of Technology and the M.Sc. and Ph.D. degrees in electrical engineering from the University of Southern California (USC), Los Angeles, in 2001 and 2006, respectively. He is currently a Professor with the Department of Communication Systems, EURECOM, Sophia Antipolis, France. His latest research deals with the intersection of coded caching and feedback-aided communications in multiuser settings. He has also worked in the area of complexity-constrained communications, MIMO, queuing theory and cross-layer design, coding theory, information theoretic limits in cooperative communications, and surveillance networks. He is a Fulbright Scholar, a co-recipient of the NEWCOM++ Distinguished Achievement Award from 2008 to 2011 for a sequence of publications on the topic of complexity in wireless communications, and a recipient of the ERC Consolidator Grant on cache-aided wireless communications from 2017 to 2022.

Antti Tölli (Senior Member, IEEE) received the Dr.Sc. (Tech.) degree in electrical engineering from the University of Oulu, Oulu, Finland, in 2008. From 1998 to 2003, he worked at Nokia Networks as a Research Engineer and the Project Manager both in Finland and Spain. In May 2014, he was granted a five year (2014–2019) Academy Research Fellow post by the Academy of Finland. During the academic year 2015 to 2016, he visited the EURECOM, Sophia Antipolis, France, and from August 2018 to June 2019, he visited the University of California at Santa Barbara, USA. He is an Associate Professor with the Centre for Wireless Communications (CWC), University of Oulu. He has authored numerous papers in peer-reviewed international journals and conferences and several patents all in the area of signal processing and wireless communications. His research interests include radio resource management and transceiver design for broadband wireless communications, with a special emphasis on distributed interference management in heterogeneous wireless networks. From 2017 to 2021, he served as an Associate Editor for IEEE TRANSACTIONS ON SIGNAL PROCESSING.