Extending the Optimality Range of Multi-Antenna Coded Caching with Shared Caches

Emanuele Parrinello*, Petros Elia* and Eleftherios Lampiris†
*Communication Systems Department, EURECOM, Sophia Antipolis, France
†Electrical Engineering and Computer Science Department, Technical University of Berlin
Email adresses: {parrinel,elia}@eurecom.fr, lampiris@tu-berlin.de

***Abstract*—This work considers the cache-aided multiple-input single-output broadcast channel (MISO BC) where an L-antenna transmitter serves K receiving users, each assisted by one of $\Lambda < K$ caches with normalized capacity γ. For this setting it was known that in the range of $L \in [K/\Lambda, K\gamma]$, one can achieve a Degrees-of-Freedom (DoF) performance of $L + K\gamma$. This very restrictive constraint that $L \leq K\gamma$ excluded shared-cache settings with a large number of antennas from achieving the maximum DoF; for the more realistic regime of $L > K\gamma$, all existing coded caching schemes suffer substantially reduced caching or multiplexing gains.**

Our work provides a novel coded caching scheme that achieves the exact best known, near optimal, DoF $L+K\gamma$, and does so even if $L > K\gamma$, thus covering an important hole in identifying the optimal performance for the multi-antenna shared-cache problem. Therefore, our work reveals that shared-cache systems with many transmit antennas can also enjoy both full multiplexing gains (L) as well as full caching gains ($K\gamma$) despite the sharing of the caches. A side benefit of this scheme is its applicability in multi-antenna settings with dedicated users caches, where it can offer the advantage of reducing the subpacketization without sacrificing the DoF performance.

I. Introduction

Coded Caching [1] was proposed as an alternative method of exploiting receiver-side caches, with its main idea being to use cacheable content as side information that alleviates interference from the receiving users. At a time of cheap storage units and abundant predictability of on-demand video traffic, coded caching is perceived as a promising tool in the effort to sustain the exponential increase of on-demand video traffic in wireless networks.

The original work in [1] considered a shared-link setting where a server hosting a library of N files, serves K receiving users, each equipped with a dedicated cache that can host a fraction $\gamma \in [0, 1]$ of the library. Assuming a link capacity of 1 file per unit of time, the work in [1] proposed an algorithm which consisted of a *cache placement phase*, during which the caches were filled with content, and a subsequent *delivery phase*, during which the demands of the K users were announced and satisfied. In this context, the work in [1] showed that any set of K simultaneous file requests can be served with worst-case (normalized) completion time of $T = \frac{K(1-\gamma)}{1+K\gamma}$, implying the ability to serve $K\gamma + 1$ users at a time; a number known as the sum *Degrees of Freedom* (DoF)

$$d_1 = \frac{K(1-\gamma)}{T} = 1 + K\gamma, \tag{1}$$

corresponding to a caching gain of $K\gamma$ additional served users due to caching. Key to achieving this gain is the ability for each user to have its own dedicated cache. This performance was later proved to be exactly optimal under the assumption of uncoded cache placement in [2], [3], and optimal within a multiplicative factor of 2.01 in [4] for the general case.

a) Multi-antenna coded caching with dedicated caches: Soon after, the work in [5] considered a similar setting, where now the server employs multiple ($L \geq 1$) transmit antennas. This work nicely showed that by adding antennas, one can achieve the DoF

$$d_L = K\gamma + L. \tag{2}$$

This reveals that the aforementioned caching gain, experienced in the single-antenna case, can be additively combined with the multiplexing gain. The DoF performance of (2) was later proved in [6] to be within a multiplicative factor of 2 from the one-shot linear optimal sum-DoF. This work sparked substantial interest in related multi-antenna coded caching settings [7], [8], many of which are motivated by the realization that coded caching stands a better chance in affecting wireless systems if it manages to work well with existing network resources, the most prominent of which being multi-antenna arrays.

b) Coded caching with shared caches: Another line of research has recently studied coded caching in the context of shared caches, where the K receiving users, instead of having their own dedicated cache, are instead served by Λ caches (cf. Fig. 1), where each cache is shared by several users. This setting is of particular interest not only because it represents more realistic wireless scenarios, where receivers (physically) share cache-aided helper nodes, but also because it reflects the effect of subpacketization constraints [9] that may force a reduced number of cache-states and thus may force cache-aided receivers to have identically-filled caches. One of the first coded caching works in the context of shared caches can be found in [10], which then motivated related works such as [11]–[13].

This work was supported by the European Research Council under the EU Horizon 2020 research and innovation program / ERC grant agreement no. 725929 and ERC grant 789190 - CARENET.

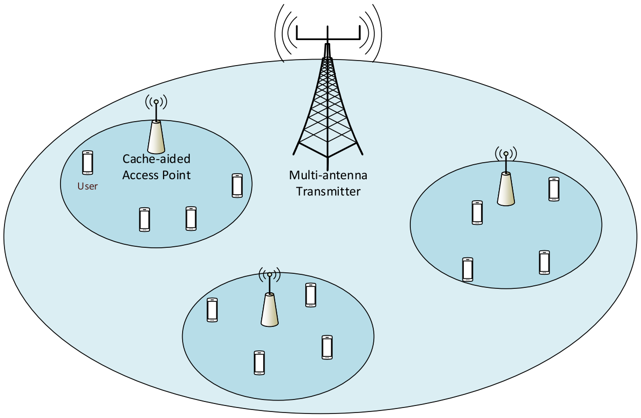

Fig. 1. The MISO BC, where each user communicates with a base station and an access point node. Each access point is equipped with an individual cache.

A. Shared caches and multiple transmit antennas

A natural continuation of this line of work involves the scenario where shared caches coexist with multi-antenna transmitters. This joint setting is again motivated not only by the expected high number of antennas in downlink antenna arrays, but also by the powerful role (cf. [14]) of multiple antennas in meeting stringent subpacketization constraints that – as suggested before – may force users to have identical cache states, i.e. which may effectively force a small Λ.

In identifying the fundamental limits of this multi-antenna shared cache setting, where an L-antenna transmitter delivers to K users with the help of Λ caches, one has to account for the fact that — while traditional coded caching methods count on each user having properly designed, and thus different, cache content — now each user is constrained to having to share the same cache content with $K/\Lambda - 1$ other users[1].

A sub-optimal solution: A naive way to account for this shared-cache constraint is to treat users with the same cache in different time-slots. For example, using the algorithm of [5] with this slotted approach, would yield a DoF of

$$d = \Lambda\gamma + L. \tag{3}$$

This approach is naturally always sub-optimal.

Optimal solutions for small number of antennas: The breakthrough came initially with the work in [14] which — in the context of subpacketization-constrained coded caching — proposed a scheme which effectively[2] employed a reduced number of Λ shared caches, to achieve the elusive DoF of $d_L = K\gamma + L$ with dramatically reduced subpacketization. However, due to the symmetric nature of the design, this scheme required that the number of antennas be limited by $L \leq K\gamma$. More precisely, this exact d_L was achieved in [14], under the constraints that

$$(i) \quad K\gamma = \alpha L \qquad (ii) \quad K = \beta L, \tag{4}$$

where α and β are integers. Removal – in [14] in the context of shared caches – of these constraints, would result in multiplicative DoF losses that could be as severe as removing a significant portion of either the multiplexing or the caching gain from the DoF. Given the conceivably modest value of γ, which in turn means a modest value of $K\gamma$, the exact performance of $d_L = L + K\gamma$ could only be achieved up to a potentially modest number of transmit antennas, thus excluding the applicability of this result from a large family of multi-antenna wireless networks. This clearly conflicts with current trends in communications that tend to employ large antenna arrays at the transmitter.

Later on, the recent work in [12] studied the fundamental limits of this same multi-antenna shared-cache setting[3], providing a novel outer bound as well as a coded caching scheme that achieved — in the current context where each cache serves an equal number of K/Λ users — an optimal DoF of

$$d = L(\Lambda\gamma + 1), \tag{5}$$

operating under the assumptions that $L \leq \frac{K}{\Lambda}$. In the scenario where $L > \frac{K}{\Lambda}$, but under the aforementioned limiting conditions in equation (4), this scheme could be modified, after further consolidating the Λ caches[4], to achieve the best known DoF $d_L = K\gamma + L$. If again $L > \frac{K}{\Lambda}$, but now without the restrictive conditions in (4), then a naive adaptation of the scheme in [12], that effectively would involve shutting down all but $\frac{K}{\Lambda}$ antennas, would yield a much reduced DoF of

$$d = \max\{K\gamma + \frac{K}{\Lambda}, L\} \tag{6}$$

resulting in the loss of a sizeable part of the multiplexing gain, from L to $\frac{K}{\Lambda}$.

To the best of our knowledge, in the shared-cache context, none of the existing schemes can achieve the exact DoF of $K\gamma + L$ when $L > K\gamma$. In this shared-caches setting, existing schemes either incur a multiplicative DoF loss (cf. [14]), or provide reduced caching gains (cf. (3)) or reduced multiplexing gains (cf. (6)).

Current Work: In this work we characterize the optimal DoF for the shared-cache multi-antenna setting where the number of antennas L exceeds the number of users per cache $\frac{K}{\Lambda}$ for the case when $L > K\gamma$. As a byproduct of the new scheme, there is the opportunity to employ our scheme in settings with dedicated caches, as a means to achieve the optimal DoF with reduced subpacketization in scenarios where L does not divide K and $K\gamma$.

[1] Here, we refer to the uniform setting where each cache serves $\frac{K}{\Lambda}$ users.

[2] In particular, the scheme involved splitting the users in $\Lambda = \frac{K}{L}$ groups such that all users in the same group store the same content.

[3] This work considered the general case where the number of users associated to each cache is arbitrary (and not necessarily always equal to K/Λ as we assume here). The optimality results in [12] come under the assumptions of uncoded cache placement, and cache placement phase that is oblivious to the number of users that will be connected to each cache during the delivery phase.

[4] For example, for $L = \Lambda = 4$, $\gamma = \frac{1}{2}$ and $K = 8$, with 2 users per cache, one could group the 4 caches into $\frac{K}{L} = 2$ groups of 2 caches each, where in each group, the caches are filled with the same content.

B. Notation

We use $\mathbb{N}, \mathbb{Z}, \mathbb{C}$ to represent the sets of natural, integer and complex numbers, respectively. For $\Lambda \in \mathbb{N}$, we use $[\Lambda] \triangleq \{1, 2, \ldots, \Lambda\}$. The expression $\alpha|\beta$ denotes that integer α divides integer β. For a set $\mathcal{A}$, $|\mathcal{A}|$ denotes its cardinality. The binomial coefficient is denoted and defined by $\binom{n}{k} \triangleq \frac{n!}{(n-k)!k!}$. We will assume that sets are ordered, and for set τ we will refer to its j-th element as $\tau(j)$.

II. System Model and Problem Formulation

We consider a caching network where a server communicates with K users via a transmitter with L antennas and has access to a library with $N \geq K$ unit-sized files $\{W^n\}_{n=1}^N$. The communication in the network is facilitated by Λ caches, each serving at *zero cost* an arbitrary set of $\frac{K}{\Lambda}$ users such that each user is served by only one cache. The memory of each cache is limited to M units of file, such that each individual cache stores only a fraction $\gamma = \frac{M}{N}$ of the library.

We assume that the system operates in 2 distinct phases.

1) *Cache placement.* During this phase, each cache $\lambda \in [\Lambda]$ is filled with content $\mathcal{Z}_\lambda$ from the library, without knowledge of future demands, and without knowledge of which user is associated to which cache.
2) *Delivery.* This phase consists of 2 steps. First, each user $k \in [K]$ requests a file W^{d_k} from the library and notifies the server with the index d_k of its demanded file. We denote by

$$\mathbf{d} = (d_1, d_2, \ldots, d_K) = \left(\mathbf{d}^1, \cdots, \mathbf{d}^\lambda, \cdots, \mathbf{d}^\Lambda\right)$$

the vector of the requested file indices, where $\mathbf{d}^\lambda$ denotes the vector of the file indices requested by users connected to cache λ. Subsequently, the server communicates a broadcast message, which is used, along with the cached content, by the users to recover their requested files.

The delivery phase consists of a set of properly designed vectors transmitted one after the other. For each transmitted vector $\mathbf{v} \in \mathbb{C}^{L \times 1}$, the received signals at user k, take the form

$$y_k = \mathbf{h}_k^T \mathbf{v} + w_k$$

where $\mathbf{h}_k \in \mathbb{C}^{L \times 1}$ denotes the channel gain of receiving user k, where $\mathbf{v}$ satisfies a power constraint $\mathbb{E}(||\mathbf{v}||^2) \leq P$, and where w_k represents the AWGN noise with unit power at receiver k. We will assume high signal-to-noise ratio SNR (high P), that the transmitter and all users have perfect channel state information (CSI), that fading is statistically symmetric, and that each link (one antenna to one receiver) has ergodic capacity $\log(SNR) + o(log(SNR))$.

The performance metric of our interest is the delivery time T, defined as the time required so that all users successfully decode their requested files. The focus of this paper is on the worst-case delivery time, i.e. the maximum delivery time required to serve any possible demand vector $\mathbf{d}$.

III. Main Results

Theorem 1. *In the L-antenna MISO BC with Λ caches, each of normalized capacity γ, K users populating the caches uniformly, then, as long as $\frac{K}{\Lambda}|L$, the order optimal performance*

$$T = \frac{K(1-\gamma)}{K\gamma + L} \tag{7}$$

is achievable.

Proof. The achievable scheme is described in Sec. IV, where the scheme works for all $\frac{K}{\Lambda}|L$, irrespective of whether $L > K\gamma$ or not. □

Example 1. Let us consider a setting with $K = 105$ users, $\Lambda = 21$ caches of normalized size $\gamma = \frac{1}{21}$ and $L = 15$ antennas. Using naively the algorithm of [5] would result in a DoF performance $d_{\text{MS}} = \Lambda\gamma + L = 16$. Given that $L \geq K\gamma$, employing the scheme in [14], by means of memory sharing, would result in the DoF performance $d_r = 15.88$. As we can see, both approaches result in a loss of most of the caching gain. In fact, the fact that $L = 15$ allows to trivially achieve DoF 15 without the use of caching. On the other hand, our scheme here achieves the DoF $d_L = K\gamma + L = 20$.

The benefits on subpacketization: Theorem 1 also applies to settings with dedicated users caches (where each user can have its own independent cache), in which case Λ serves as a design parameter that regulates the subpacketization requirement of the scheme. For this setting with dedicated caches, the recent work in [16] proposed a new algorithm for the case $L \geq K\gamma$, which achieves the optimal DoF $d_L = K\gamma + L$ with a subpacketization requirement of $K \cdot (K\gamma + L)$. In this regime, the algorithm in [16] removes the memory-sharing requirement in [14], thus improving the DoF at a cost of a reasonable subpacketization increase, from approximately a linear K/L in [14] to the quadratic $K \cdot (K\gamma + L)$. For this same regime where $L \geq K\gamma$, our new scheme here can also achieve the same DoF $d_L = K\gamma + L$, provided that there exist a Λ such that $\frac{K}{\Lambda}|L$ and $\Lambda\gamma \in \mathbb{N}$. As we will describe later on in this section, our scheme requires a subpacketization of $S_{new} = \binom{\Lambda}{\Lambda\gamma} \cdot \binom{\Lambda-\Lambda\gamma-1}{\frac{L\Lambda}{K}-1}$, which can be smaller – for some setting parameters – than the subpacketization $S = K \cdot (K\gamma + L)$ required by [16].

Example 2. Consider a setting with $K = 80$ users, $L = 30$ antennas and a per-user normalized cache size $\gamma = \frac{1}{8}$. Both the scheme in [16] and our proposed algorithm achieve the DoF of 40; however while the first requires to split each file into $S = 3200$ subfiles, our scheme necessitates a subpacketization of only $S_{new} = 120$ (by setting $\Lambda = 8$), which is approximately 27 times smaller. For completeness, we note that the scheme in [14] with memory-sharing would result in a mere subpacketization of $S = 8$, but with a much reduced DoF of 34.3.

Remark 1. Although the results require that L is a multiple of $\frac{K}{\Lambda}$, the optimal DoF of $K\gamma + L$ can in fact be achieved in the general case of $L \geq \frac{K}{\Lambda}$ by modifying the scheme described in Section IV. The difference is that, while in our current scheme each transmitted vector contains subfiles for all users associated to a chosen subset of caches, in the modified scheme each transmission serves fully all users of some caches and only a subset of users of some other caches. The description of this modified version of the scheme will appear in a future extended version of this current work.

IV. Scheme Description

In this section we will describe the algorithm that achieves the result of Theorem 1. We begin with describing the algorithm through the use of an example and continue with the general algorithm.

A. Example

Consider a scenario where a base station equipped with $L = 4$ antennas has access to a library with $N = 8$ files $W^1, W^2, \dots, W^8$, and is connected to $K = 8$ users. We assume that the number of caches is $\Lambda = 4$ and that users are distributed uniformly among the caches, such that each cache serves $\frac{K}{\Lambda} = 2$ users. Furthermore, we also assume that the per-cache capacity is $M = 2$ (i.e., $\gamma = \frac{1}{4}$).

In the cache placement phase, each file $W^n, n \in [8]$ is split into $\binom{\Lambda}{\Lambda\gamma} = 4$ equally-sized subfiles denoted by $W^n_1, W^n_2, W^n_3, W^n_4$. Employing the cache placement algorithm of [1], we have that each cache $\lambda \in [4]$ stores $W^n_\lambda, \forall n \in [8]$. In the delivery phase, we further split each subfile W^n_τ, $\tau \in [4]$ into 2 equally-sized and disjoint minifiles $W^n_{\tau,1}, W^n_{\tau,2}$. For simplicity, we will also use the notation $A \equiv W^1$, $B \equiv W^2$, $C \equiv W^3$, and so on.

Without loss of generality, we assume that users $\{1, 5\}$ are connected to the first cache and request files $\{A, E\}$, users $\{2, 6\}$ are connected to the second cache and request $\{B, F\}$, and so on.

For simplicity, in this example we will use the *one-shot* (see Section IV-E) variation of the proposed scheme in Section IV-C. The delivery algorithm consists of 4 rounds, each serving users from $\Lambda\gamma + \frac{L\Lambda}{K} = 3$ different caches.

In the first round, the server transmits to users in caches $1, 2, 3$ the vector

$$\mathbf{v}_{1,2,3} = \mathbf{H}^{-1}_{1,5,3,7}\begin{bmatrix} A_{2,1} \\ E_{2,1} \\ C_{2,1} \\ G_{2,1}\end{bmatrix} + \mathbf{H}^{-1}_{1,5,2,6}\begin{bmatrix} A_{3,1} \\ E_{3,1} \\ B_{3,1} \\ F_{3,1}\end{bmatrix} + \mathbf{H}^{-1}_{2,6,3,7}\begin{bmatrix} B_{1,1} \\ F_{1,1} \\ C_{1,1} \\ G_{1,1}\end{bmatrix}$$

in a time slot of normalized duration $\frac{2}{8}$. $\mathbf{H}^{-1}_{i,j,p,q}$ is the zero-forcing (ZF) precoder that inverts the channel matrix $\mathbf{H}_{i,j,p,q} \triangleq [\mathbf{h}_i^T \mathbf{h}_j^T \mathbf{h}_p^T \mathbf{h}_q^T]$. To describe the decoding, we focus on user 1 who receives the signal

$$y_1 = A_{2,1} + A_{3,1} + \underbrace{\mathbf{h}_1^T \cdot \mathbf{H}^{-1}_{2,6,3,7}\begin{bmatrix} B_{1,1} \\ F_{1,1} \\ C_{1,1} \\ G_{1,1}\end{bmatrix}}_{\text{interference}} + w_1. \quad (8)$$

We observe that user 1 is connected to cache 1 and that it has perfect knowledge of the channel state, thus user 1 can reconstruct the interference term in (8) and subtract it from y_1 to obtain (neglecting the noise) $\bar{y}_1 = A_{2,1} + A_{3,1}$. Recalling that $|A_{2,1}| = |A_{3,1}| = \frac{1}{8}$, which is half of the transmission duration for vector $\mathbf{v}_{1,2,3}$, user 1 can successfully decode the 2 desired minifiles $A_{2,1}, A_{3,1}$. The same decoding procedure is applied to the other users served in the first round.

Similarly, in the other 3 rounds the transmitted vectors are

$$\mathbf{v}_{1,2,4} = \mathbf{H}^{-1}_{1,5,4,8}\begin{bmatrix} A_{2,2} \\ E_{2,2} \\ D_{2,1} \\ H_{2,1}\end{bmatrix} + \mathbf{H}^{-1}_{1,5,2,6}\begin{bmatrix} A_{4,1} \\ E_{4,1} \\ B_{4,1} \\ F_{4,1}\end{bmatrix} + \mathbf{H}^{-1}_{2,6,4,8}\begin{bmatrix} B_{1,2} \\ F_{1,2} \\ D_{1,1} \\ H_{1,1}\end{bmatrix},$$

$$\mathbf{v}_{1,3,4} = \mathbf{H}^{-1}_{1,5,4,8}\begin{bmatrix} A_{3,2} \\ E_{3,2} \\ D_{3,1} \\ H_{3,1}\end{bmatrix} + \mathbf{H}^{-1}_{1,5,3,7}\begin{bmatrix} A_{4,2} \\ E_{4,2} \\ C_{4,1} \\ G_{4,1}\end{bmatrix} + \mathbf{H}^{-1}_{4,8,3,7}\begin{bmatrix} D_{1,2} \\ H_{1,2} \\ C_{1,2} \\ G_{1,2}\end{bmatrix},$$

$$\mathbf{v}_{2,3,4} = \mathbf{H}^{-1}_{2,6,4,8}\begin{bmatrix} B_{3,2} \\ F_{3,2} \\ D_{3,2} \\ H_{3,2}\end{bmatrix} + \mathbf{H}^{-1}_{2,6,3,7}\begin{bmatrix} B_{4,2} \\ F_{4,2} \\ C_{4,2} \\ G_{4,2}\end{bmatrix} + \mathbf{H}^{-1}_{3,7,4,8}\begin{bmatrix} C_{2,2} \\ G_{2,2} \\ D_{2,2} \\ H_{2,2}\end{bmatrix}.$$

The overall optimal delivery time required to serve all users' demands is $T = \frac{2}{8} \cdot 4 = 1$, which corresponds to a sum degrees of freedom of $DoF = \frac{8(1-1/4)}{T} = 6 = K\gamma + L$.

We proceed with the description of the general scheme.

B. Cache Placement Scheme

The cache placement phase is the same as the one in [1], for a setting with Λ users, each with its own dedicated cache. Therefore, each file $W^n, n \in [N]$ is split into $\binom{\Lambda}{\Lambda\gamma}$ disjoint subfiles W^n_τ, for each $\tau \subset [\Lambda], |\tau| = \Lambda\gamma$. Then, each cache λ stores a fraction γ of the library according to the following policy

$$\mathcal{Z}_\lambda = \{W^n_\tau : \tau \ni \lambda, \forall n \in [N]\}. \quad (9)$$

C. Delivery Scheme

Upon receiving the users' requests, the server further splits the demanded subfiles $W^{d_k}_\tau$ in $\binom{\Lambda-\Lambda\gamma-1}{\frac{L\Lambda}{K}-1}$ minifiles as follows

$$W^n_\tau = \left\{ W^n_{\tau,r} : r \in \left\{1, 2, \dots, \binom{\Lambda - \Lambda\gamma - 1}{\frac{L\Lambda}{K} - 1}\right\}\right\}. \quad (10)$$

For each set of caches $\Phi \subseteq [\Lambda]$, each of cardinality $|\Phi| = \Lambda\gamma + \frac{L\Lambda}{K}$, the server transmits $\binom{\Lambda\gamma + \frac{L\Lambda}{K} - 1}{\Lambda\gamma}$ vectors of the form

$$\mathbf{v}^{(i)}_\Phi = \sum_{\phi \subset \Phi : |\phi| = \Lambda\gamma} c^{(i)}_{\Phi_{\backslash\phi}} \cdot \mathbf{H}^{-1}_{\Phi_{\backslash\phi}} \cdot \begin{bmatrix} \mathbf{W}^{\mathbf{d}^{\Phi\backslash\phi(1)}}_{\phi,r_1} \\ \mathbf{W}^{\mathbf{d}^{\Phi\backslash\phi(2)}}_{\phi,r_2} \\ \vdots \\ \mathbf{W}^{\mathbf{d}^{\Phi\backslash\phi\left(\frac{L\Lambda}{K}\right)}}_{\phi,r_{\frac{L\Lambda}{K}}} \end{bmatrix} \quad (11)$$

for $i \in \left\{1, 2, \dots, \binom{\Lambda\gamma + \frac{L\Lambda}{K} - 1}{\Lambda\gamma}\right\}$. In the above, $c^{(i)}_{\Phi_{\backslash\phi}} \in \mathbb{C}$ denotes an arbitrary coefficient, $\Phi_{\backslash\phi}(l), l \in \left[\frac{L\Lambda}{K}\right]$ denotes the l-th element of the ordered set of caches $\Phi\backslash\phi$, where $\phi \subset \Phi : |\phi| = \Lambda\gamma$, and $\mathbf{W}^{\mathbf{d}^{\Phi\backslash\phi(l)}}_{\phi,r_l}$ denotes a $\frac{K}{\Lambda} \times 1$ vector of minifiles requested by all users connected to cache $\Phi_{\backslash\phi}(l)$, i.e.

$$\mathbf{W}_{\phi,r_l}^{\mathbf{d}^{\Phi\backslash\phi(l)}} \triangleq \begin{bmatrix} W_{\phi,r_l}^{\mathbf{d}^{\Phi\backslash\phi(l)}(1)} \\ W_{\phi,r_l}^{\mathbf{d}^{\Phi\backslash\phi(l)}(2)} \\ \vdots \\ W_{\phi,r_l}^{\mathbf{d}^{\Phi\backslash\phi(l)}(\frac{K}{\Lambda})} \end{bmatrix}.$$

The choice of indices $r_1, r_2, \ldots, r_{\frac{L\Lambda}{K}}$ is sequential, guaranteeing that no minifile is transmitted twice.

a) Decoding: Directly from (11) and for a fixed Φ and i, the received signal at the $q-$th user, denoted by u, of cache $\lambda \in \Phi$ is

$$y_{u,\Phi}^{(i)} = \underbrace{\sum_{\phi\subset\Phi\backslash\{\lambda\}:|\phi|=\Lambda\gamma} c_{\Phi_{\backslash\phi}}^{(i)} W_{\phi,r}^{d_u}}_{\mathcal{L}_{\Phi,u}^{(i)}} + \iota_u^{(i)}$$

where $\mathcal{L}_{\Phi,u}^{(i)}$ is the part of the received signal useful for user u, while $\iota_u^{(i)}$ is an interference term that takes the form

$$\mathbf{h}_u^T \cdot \sum_{\phi\subset\Phi:\phi\ni\lambda,|\phi|=\Lambda\gamma} c_{\Phi_{\backslash\phi}}^{(i)} \cdot \mathbf{H}_{\Phi_{\backslash\phi}}^{-1} \cdot \begin{bmatrix} \mathbf{W}_{\phi,r_1}^{\mathbf{d}^{\Phi\backslash\phi(1)}} \\ \mathbf{W}_{\phi,r_2}^{\mathbf{d}^{\Phi\backslash\phi(2)}} \\ \vdots \\ \mathbf{W}_{\phi,r_{\frac{L\Lambda}{K}}}^{\mathbf{d}^{\Phi\backslash\phi(\frac{L\Lambda}{K})}} \end{bmatrix} + w_u.$$

We can see that user u can reconstruct and remove (neglecting the noise) the interference term $\iota_u^{(i)}$, because it can obtain the minifiles contained in $\iota_u^{(i)}$ through its cache. After collecting the signals

$$\bar{y}_{u,\Phi}^{(i)} = y_{u,\Phi}^{(i)} - \iota_u^{(i)}, \qquad \forall i \in \left[\binom{\Lambda\gamma + \frac{L\Lambda}{K} - 1}{\Lambda\gamma}\right],$$

user u will possess $\binom{\Lambda\gamma+\frac{L\Lambda}{K}-1}{\Lambda\gamma}$ linear combinations $\mathcal{L}_{\Phi,u}^{(i)}$ of the same set of desired minifiles $\Psi = \left\{W_{\phi,r_l}^u | \phi \in \Phi \setminus \{g\}, |\phi| = \Lambda\gamma\right\}$.

Directly from (11), we observe that $\mathbf{v}_\Phi^{(i)}$ is constructed as the sum of $\binom{\Lambda\gamma+\frac{L\Lambda}{K}}{\Lambda\gamma}$ vectors, each containing minifiles for users connected to the caches in the set $\Phi \setminus \phi$, where $\phi \subset \Phi : |\phi| = \Lambda\gamma$. Taking this into account, the number of times that a cache $\lambda \in \Phi$ appears in all the sets $\Phi \setminus \phi$ appearing in $\mathbf{v}_\Phi^{(i)}$ can be computed to be $\binom{\Lambda\gamma+\frac{L\Lambda}{K}-1}{\Lambda\gamma}$. This means that the vector $\mathbf{v}_\Phi^{(i)}$ contains $|\Psi| = \binom{\Lambda\gamma+\frac{L\Lambda}{K}-1}{\Lambda\gamma}$ minifiles for each user in cache λ.

As a result, user u can successfully decode all the desired files in Ψ from the $\binom{\Lambda\gamma+\frac{L\Lambda}{K}-1}{\Lambda\gamma}$ linear combinations $\mathcal{L}_{\Phi,u}^{(i)}$.

D. Performance of the scheme

We observe that the scheme splits each file into $S = \binom{\Lambda}{\Lambda\gamma}\binom{\Lambda-\Lambda\gamma-1}{\frac{L\Lambda}{K}-1}$ minifiles, where the first term follows from the subpacketization required by the cache placement in equation (9) and the second term is due to the further subpacketization needed by the delivery phase (cf. (10)).

Direcly from the scheme, the total number of transmissions can be easily computed as $\mathcal{N} = \binom{\Lambda}{\Lambda\gamma+\frac{L\Lambda}{K}}\binom{\Lambda\gamma+\frac{L\Lambda}{K}-1}{\Lambda\gamma}$. Finally, we notice that all transmissions have the same duration of $\frac{1}{S}$, yielding a total delivery time of

$$T = \frac{\binom{\Lambda}{\Lambda\gamma+\frac{L\Lambda}{K}}\binom{\Lambda\gamma+\frac{L\Lambda}{K}-1}{\Lambda\gamma}}{\binom{\Lambda}{\Lambda\gamma}\binom{\Lambda-\Lambda\gamma-1}{\frac{L\Lambda}{K}-1}} = \frac{K(1-\gamma)}{K\gamma+L}.$$

E. A one-shot linear variation of the delivery scheme

In this subsection, we present a variation of the delivery phase presented in Section IV-C. Unlike the previous algorithm, this scheme has the *one-shot* property, where each part of the requested messages is transmitted only once. This property allows us to apply the optimality result in [6].

In the new scheme, each transmission associated to Φ occurs in a time slot of duration $T_s = \frac{\binom{\Lambda\gamma+\frac{L\Lambda}{K}-1}{\Lambda\gamma}}{S}$, where the server transmits the message

$$\mathbf{v}_\Phi = \sum_{\phi\subset\Phi:|\phi|=\Lambda\gamma} \mathbf{H}_{\Phi_{\backslash\phi}}^{-1} \cdot \begin{bmatrix} \mathbf{W}_{\phi,r_1}^{\mathbf{d}^{\Phi\backslash\phi(1)}} \\ \mathbf{W}_{\phi,r_2}^{\mathbf{d}^{\Phi\backslash\phi(2)}} \\ \vdots \\ \mathbf{W}_{\phi,r_{\frac{L\Lambda}{K}}}^{\mathbf{d}^{\Phi\backslash\phi(\frac{L\Lambda}{K})}} \end{bmatrix} \tag{12}$$

of duration $|\mathbf{v}_\Phi| = \frac{1}{S}$. After receiving the signal $\mathbf{h}_u^T\mathbf{v}_\Phi$, user u removes the interference term as described in Section IV-C. This step presents user u with a multiple access channel (MAC) with $|\Psi| = \binom{\Lambda\gamma+\frac{L\Lambda}{K}-1}{\Lambda\gamma}$ messages to resolve. Having the slot duration T_s be $\binom{\Lambda\gamma+\frac{L\Lambda}{K}-1}{\Lambda\gamma}$ times larger than the message duration $|\mathbf{v}_\Phi|$, guarantees that we are within the achievable rate region of the MAC. The rest of the calculations follow as before.

V. Conclusions

We presented a new coded caching scheme for the multi-antenna shared-caches setting that achieves the exact best known, near optimal DoF $L+K\gamma$, without the limiting condition that the number of antennas be bounded by $L < K\gamma$. This now tells us that systems with many transmit antennas, can also enjoy both full multiplexing gains as well as full caching gains associated to having dedicated (non-shared) caches. In addition to the existing result that for $L \leq K/\Lambda$ the optimal DoF (under the assumption of uncoded cache placement) is $L(1+\Lambda\gamma)$, we now know that for $L \geq K/\Lambda$, the DoF of $L+K\gamma$ is achievable (and optimal within a factor of 2, under the assumption of one-shot linear schemes [6]). For increased numbers of antennas and users, the new scheme allows for substantially increased caching or multiplexing gains, and – for some parameters – allows for subpacketization reductions over the state of art.

References

[1] M. Maddah-Ali and U. Niesen, "Fundamental limits of caching," *IEEE Trans. Inf. Theory*, May 2014.

[2] K. Wan, D. Tuninetti, and P. Piantanida, "On the optimality of uncoded cache placement," in *Information Theory Workshop (ITW), IEEE*, 2016.

[3] Q. Yu, M. A. Maddah-Ali, and A. S. Avestimehr, "The exact rate-memory tradeoff for caching with uncoded prefetching," *IEEE Trans. Inf. Theory*, Feb 2017.

[4] 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*, Jan 2019.

[5] S. P. Shariatpanahi, S. A. Motahari, and B. H. Khalaj, "Multi-server coded caching," *IEEE Trans. Inf. Theory*, Dec 2016.

[6] N. Naderializadeh, M. A. Maddah-Ali, and A. S. Avestimehr, "Fundamental limits of cache-aided interference management," *IEEE Trans. Inf. Theory*, May 2017.

[7] S. P. Shariatpanahi, G. Caire, and B. H. Khalaj, "Physical-layer schemes for wireless coded caching," *IEEE Trans. Inf. Theory*, 2018.

[8] A. Tölli, S. P. Shariatpanahi, J. Kaleva, and B. Khalaj, "Multicast beamformer design for coded caching," in *Proc. IEEE Int. Symp. on Inform. Theory (ISIT)*, June 2018.

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

[10] J. Hachem, N. Karamchandani, and S. Diggavi, "Coded caching for multi-level popularity and access," *IEEE Trans. Inf. Theory*, May 2017.

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

[12] E. Parrinello, A. Ünsal, and P. Elia, "Fundamental limits of coded caching with multiple antennas, shared caches and uncoded prefetching," *IEEE Trans. Inf. Theory*, 2019.

[13] K. Wan, D. Tuninetti, M. Ji, and G. Caire, "A novel cache-aided fog-ran architecture," in *Proc. IEEE Int. Symp. on Inform. Theory (ISIT)*, July 2019.

[14] E. Lampiris and P. Elia, "Adding transmitters dramatically boosts coded-caching gains for finite file sizes," *IEEE Journal on Selected Areas in Communications (Special Issue on Caching)*, June 2018.

[15] N. S. Karat, S. Dey, A. Thomas, and B. S. Rajan, "An optimal linear error correcting delivery scheme for coded caching with shared caches," 2019. [Online]. Available: http://arxiv.org/abs/1901.03188

[16] M. Salehi, A. Tölli, and S. Shariatpanahi, "A multi-antenna coded caching scheme with linear subpacketization," 2019. [Online]. Available: https://arxiv.org/abs/1910.10384