# Bridging two extremes: Multi-antenna Coded Caching with Reduced Subpacketization and CSIT

Eleftherios Lampiris
EURECOM
Sophia Antipolis, France
Email: lampiris@eurecom.fr

Petros Elia
EURECOM
Sophia Antipolis, France
Email: elia@eurecom.fr

***Abstract*— Two long-standing bottlenecks of coded caching are the exponentially large file sizes that are needed to achieve a maximal caching gain, and — when multiple transmit antennas are involved — the large CSIT feedback costs needed for such caching gains to materialize. Recent results have addressed these two bottlenecks individually, allowing significant reductions in both. In the Multiple-Input-Single-Output (MISO) BC with $L$ antennas, the subpacketization constraint was shown to be as small as the $L$-th root of the single-antenna case, while maintaining the full Degrees-of-Freedom performance, but requiring feedback from all active users. On the other hand, another result showed that the feedback cost can be reduced to feedback from only $L$ users, but that method required a very high subpacketization.**

**In this work we make progress towards combining the advantages of both worlds by proposing a near-optimal multi-antenna coded caching algorithm that incurs a minimal feedback cost, untangled from the number of users, and simultaneously achieving an exponentially reduced subpacketization compared to the single-stream case. In particular — in the context of the $L$-antenna MISO BC with $K$ single-antenna receivers equipped with caches of normalized size $\gamma$ — the new algorithm achieves a DoF of $(L+K\gamma)(1-\gamma)$ using feedback from only $L$ users, while also reducing subpacketization by a multiplicative factor of $\left(\frac{L+K\gamma}{1+K\gamma}\right)^{K\gamma}$ compared to the single-antenna case, achieving subpacketization reductions exponential to the number of antennas.**

## I. Introduction

The *Coded Caching* work of Maddah-Ali and Niesen [1] considered a noise-less (bottleneck) broadcast channel comprized of a server and $K$ cache-aided receivers, and showed that the caching phase can be designed in a way that can allow for multicasting opportunities during the request period. For a setting where the server has access to a library of $N$ files, where each receiver can cache the equivalent of $M$ files (thus having a normalized cache size of $\gamma \triangleq \frac{M}{N}$), and where each receiver eventually (and simultaneously) requests one library file, the authors in [1] designed a caching algorithm and a delivery algorithm such that, under the worst case demand where each user requests a different file, the normalized delivery time takes the form $T_1 = \frac{K(1-\gamma)}{1+K\gamma}$ corresponding to a degrees-of-freedom[1] (DoF) performance $D_1 = \frac{K(1-\gamma)}{T_1} = 1+K\gamma$, which was shown in [2] to be at most a multiplicative factor of 2 from optimal, and which was shown in [3] to be exactly optimal under the assumption of uncoded placement.

### *Coded Caching and Multiple Antennas*

Subsequent works (see [4]–[8]) have applied the ideas of [1] to multi-antenna fully-connected channels with the purpose of combining the multicasting gain, attributed to caching, with the multiplexing gain attributed to having multiple antennas. In the setting of the $L$-antenna Multiple-Input-Single-Output (MISO) Broadcast Channel (BC) and its DoF-equivalent Interference Channel, with $K_T$ transmitters collectively storin the whole library $L$ times, i.e. each transmitter has normalized cache $\gamma_T = \frac{L}{K_T}$, with $K$ single-antenna users, each equipped with a cache of normalized size $\gamma$, the work in [4] showed the achievability of

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

corresponding to a DoF of $D_L = L + K\gamma$, which was shown in [5] to incur a multiplicative gap from the optimal of at most 2, under the assumption of linear and one-shot transmission schemes.

### *Coded Caching and Subpacketization*

Despite this impressive theoretical performance, the above coded caching gains required that each file be divided into an exponentially large number of packets (subpackets). For example, in the single stream setting ($L = 1$), this number was of the form

$$S_1 = \binom{K}{K\gamma} \geq \left(\frac{1}{\gamma}\right)^{K\gamma} \tag{2}$$

which implied a severely deteriorated performance of coded caching (CC) in practical *subpacketization constrained* scenarios where the maximum number of packets is upper bounded by practical factors such as the file-size[2]. This subpacketization bottleneck — which constitutes a long standing problem in

The work is supported by the European Research Council under the EU Horizon 2020 research and innovation program / ERC grant agreement no. 725929. (ERC project DUALITY).

[1]The DoF express the delivery rate at high Signal-to-Noise-Ratio SNR (in units of *file*, after normalising by $\log(\text{SNR})$) and reflect the total number of users served per-transmission slot.

[2]For a more detailed exposition of the subpacketization constraint and its impact on coded caching gains see [8].

CC literature (see [9]–[15]) — was further exacerbated in the multi-antenna setting where algorithms that sought to exploit both multicasting and multiplexing gains, required even higher subpacketization that further increased exponentially as the number of antennas became larger.

In a recent development, for the same MISO BC setting and the equivalent Interference Channel setting, the work in [8] proposed an alternative way of using multiple antennas, which managed to severely ameliorate the high subpacketization problem, and while achieving the same multi-antenna performance as Eq. (1), it allowed for subpacketization that is approximately equal to the $L$-th root of the single antenna case. In practice, this reduced subpacketization allows for up to an $L$-fold increase in the subpacketization-constrained DoF performance of the single antenna case (for a more detailed analysis, the reader is referred to [8]).

*The feedback bottleneck of Multi-antenna CC*

Another bottleneck of several multi-antenna coded caching algorithms relates to feedback. This bottleneck stemmed from the requirement to have $C = L + K\gamma$ Channel State Information (CSI) training slots in the uplink and $C = L+K\gamma$ training slots in the downlink, which corresponds to a feedback cost equal to the DoF, and which thus increases with $K$.

For example, in the work of [4] a transmitted vector corresponding to an $L = 2$ antenna system, with $K = 4$ users each having a cache of size $\gamma = \frac{2}{4}$, takes the form

$$\begin{aligned}\mathbf{x} =&\mathbf{h}_1^{-1}B_{34}C_{24}D_{23} + \mathbf{h}_2^{-1}A_{34}C_{14}D_{13}+\\ &+\mathbf{h}_3^{-1}A_{24}B_{14}D_{12} + \mathbf{h}_4^{-1}A_{23}B_{13}C_{12}\end{aligned} \tag{3}$$

where $\mathbf{h}_\pi^{-1}$ is a vector orthogonal to the channels of all the users in set $\pi$. The above reveals that the CSI for *all* users should be known at both the transmitter and receivers, which in turn incurs reduced performance due to delays incurred by CSI training (cf. [16]).

This performance bottleneck was highlighted in [6], [7], which proceeded to substantially ameliorate it by providing an algorithm that achieves the full DoF (Eq. (1)), using only $C = L$ training slots (in each of the uplink and downlink training phases), but did so by requiring very high subpacketization.

*The Subpacketization/CSIT Conundrum and Results Overview*

In the above, we can discern two antipodal approaches, with the first giving very low subpacketization [8] at very high feedback costs, and with the second having very low CSIT requirements but with extremely high subpacketization. It is thus natural to ask the question of whether subpacketization reductions can coincide with CSI reductions.

*Outline of the results and general methodology:* In this work we will achieve a reduced subpacketization of $L_c\binom{K/L_c}{K\gamma}$ for $L_c = \frac{L+K\gamma}{1+K\gamma}$, and we will do so with feedback cost $C = L$ and an achieved DoF of $D = (L + K\gamma)(1 - \gamma)$.

To do so, we will manipulate caching in order to effectively 'shift' cache capacity across the users[3], thus creating an asymmetry that effectively splits the users into a group that enjoys very large caches, and another group that is effectively cache-less. This trick will then allow us to exploit the surprising new finding in [17] that, in such hybrid settings where cache-aided users coincide with cache-less users, all users (even cache-less users) can benefit from full multiplexing as well as *full caching gains*. This will in turn manifest itself into a sequence of reduced dimensionality problems with subsequent CSI and subpacketization benefits.

[3]We clarify that each user has the same actual cache size, and that no communication takes place between the users.

## II. System Model and Notation

We consider the $L$-antenna fully-connected MISO BC with $K$ single-antenna receivers. The transmitter has access to a library of $N$ files $\{W^n\}_{n=1}^N$, each of size $f$ bits, while it is assumed that the receivers will ask for one of those files. Each receiver is endowed with a cache able to store the equivalent of $M < N$ files, corresponding to a normalized cache size $\gamma \triangleq \frac{M}{N}$. Transmission takes place in two distinct phases. First, during the *pre-fetching phase*, the caches of the users are filled with content from the library, without knowledge of future file requests. Then, during the subsequent *delivery phase*, each user $k \in \{1, 2, ..., K\} \triangleq [K]$ requests file $W^{r_k}$, $r_k \in [N]$ and the transmitter takes into account the stored content at each user and transmits a vector message to satisfy these demands. A message received at user $k \in [K]$ takes the form

$$y(k) = \mathbf{h}_k^T\mathbf{x} + w_k = \sum_{i=1}^{L} h_{i,k}x_i + w_k \tag{4}$$

where $\mathbf{h}_k^T \triangleq [h_{1,k}, ..., h_{L,k}] \in \mathbb{C}^{1\times L}$ represents the channel vector from the $L$ antenna transmitter to user $k$, where $\mathbf{x}$ represents the transmitted vector satisfying some power constraint, and where $w_k \sim \mathbb{C}\mathcal{N}(0, 1)$ represents the noise experienced at user $k$.

*Notation:* For a set $A$, we will use $|A|$ to indicate its cardinality, and for sets $A, B$ we use $A\backslash B$ to denote the difference set. $\oplus$ denotes the bit-wise XOR operation, while for $a, b \in \{1, 2, ...\}$ and $a \geq b$, symbol $\binom{a}{b}$ denotes the binomial coefficient. In a small abuse of notation, we will sometimes denote transmitted messages the same way we denote the data that these transmitted messages convey. We will denote with $\mathcal{H}_\lambda^{-1}$, the $L \times L$ normalized inverse matrix of the channel matrix between the $L$-antenna transmitter and the $L$ users in set $\lambda \subset [K]$. In particular, $\mathcal{H}_\lambda^{-1} \triangleq [\mathbf{h}_{\lambda\backslash\{1\}}^{-1}, ...\mathbf{h}_{\lambda\backslash\{L\}}^{-1}]$ where

$$\mathbf{h}_k^T\mathbf{h}_{\lambda\backslash\{i\}}^{-1} = \begin{cases}0, & k \in \lambda \setminus \{i\}\\ \neq 0, & k \notin \lambda \setminus \{i\}.\end{cases} \tag{5}$$

*Feedback and Precoding:* We assume that feedback is perfect and instantaneous, while the feedback training process is divided into 2 phases; the uplink phase where the transmitter estimates the channels of some users, and the downlink phase where receiver $k$ estimates products $\mathbf{h}_k^T\mathbf{h}_\lambda^{-1}$, for some set $\lambda \subset [K], |\lambda| = L$. The feedback training process follows that of [7], hence feedback for $C$ users will require $C$ training slots in the uplink training phase (for CSIT) and $C$ training slots in the downlink training phase (for local and global CSIR).

For simplicity, we will assume that the selected precoder is a Zero-Forcing (ZF) precoder, which is adequate for a DoF analysis. Finally, without loss of generality[4], we will assume that $K$ is an integer multiple of $L+K\gamma$ and, also, that $L+K\gamma$ is an integer multiple of $1+K\gamma$.

## III. Main Results

**Theorem 1.** *In the L-antenna MISO BC with $K$ single-antenna users, each equipped with a cache of normalized size $\gamma$, the DoF of $D_L = (1-\gamma)(L+K\gamma)$ is achievable with per-transmission CSI cost $C = L$ and subpacketization of*

$$S = L_c \binom{\frac{K}{L_c}}{K\gamma}, \quad \textit{where } L_c = \frac{L+K\gamma}{1+K\gamma}. \tag{6}$$

*Proof.* The proof is constructive and described in Sec. V. □

**Remark 1.** *We observe that, as the number of antennas increases, the subpacketization reduction (with respect to the single-stream case[5] (2)) can be very substantial. For example, if $L = K\gamma + 2$, the subpacketization approximately reduces by a (multiplicative) factor of $S_r = 2^{K\gamma}$.*

*Furthermore, in the limit of asymptotically large $K$, and for $L \gg 1$, we see that the multiplicative reduction in subpacketization takes the form*

$$\lim_{K\to\infty} S_{\mathcal{R}} = \lim_{K\to\infty} \left(\frac{L+K\gamma}{1+K\gamma}\right)^{K\gamma} = \lim_{K\to\infty} \left(\frac{L+K\gamma}{K\gamma}\right)^{K\gamma}$$
$$= \lim_{K\to\infty} \left(\frac{L/\gamma}{K}+1\right)^{K\gamma} = e^{\frac{L}{\gamma}\gamma} = e^L \tag{7}$$

*implying that every additional antenna, in addition to increasing the DoF, also reduces subpacketization by a factor of $e$.*

*Finally, it is easy to conclude that for $\gamma \le \frac{1}{2}$, the achieved performance is within a factor of 4 from the one-shot linear optimal DoF.*

**Remark 2.** *As we will see later on, part of the algorithm uses the generation of XORs of [1]. In order to achieve an even smaller subpacketization, this part of our scheme can be substituted for one of the approaches of [10]–[12], by incuring a small DoF reduction.*

## IV. Scheme Description

*Scheme Intuition:* The main idea borrows from the result of [17], where it was shown that the DoF of an $L$-antenna MISO BC system with $K_c$ cache-aided users ($\gamma_c > 0$) and $K_n$ non-cache-aided[6] ($\gamma_n = 0$) users is equal to $D_L^{\text{het}} = L + K_c\gamma_c$, which exactly matches the DoF (and the delay) of an equivalent homogeneous system with $K = K_c + K_n$ users each equipped with a cache of normalized size $\gamma = \frac{K_c\gamma_c}{K_n+K_c}$.

In this work, we exploit this idea of having 'cache-less' users that benefit from full caching gains, in order to achieve a reduction in the problem dimensionality by a factor of $L_c$. This will be achieved by partitioning each file into $L_c$ parts and then by grouping the users into $L_c$ groups, where each group of $K/L_c$ users will cache content that is exclusively from just one of the $L_c$ parts of the library content. Hence for each such part, the associated group of $\frac{K}{L_c}$ users will be 'forced' to store each file with a large redundancy $K\gamma$, while simultaneously all remaining $K - \frac{K}{L_c}$ users will not cache this part at all. This further means that for each particular part, there is a group of $\frac{K}{L_c}$ cache-aided users, and the rest can be considered to be cache-less.

We proceed with the placement and delivery phases.

### A. Placement Phase

We start by dividing the users into $L_c = \frac{L+K\gamma}{1+K\gamma}$ groups

$$\mathcal{K}_1 = \left\{1, ..., \frac{K}{L_c}\right\}, ..., \mathcal{K}_{L_c} = \left\{(L_c-1)\frac{K}{L_c}+1, ..., K\right\}. \tag{8}$$

Then we split each file $W^n, n \in [N]$ into $L_c = \frac{L+K\gamma}{1+K\gamma}$ parts $\{W_1^n, ..., W_{L_c}^n\}$, and further each part into $\binom{\frac{K}{L_c}}{K\gamma}$ subfiles i.e.,

$$W_p^n \to \left\{W_{p,\tau}^n,\ \tau \subset \mathcal{K}_p,\ |\tau| = K\gamma\right\}, p \in [L_c] \tag{9}$$

hence, a total subpacketization level of $S = L_c \cdot \binom{\frac{K}{L_c}}{K\gamma}$.

Caching at user $k_p \in \mathcal{K}_p, p \in [L_c]$ takes the form

$$Z_{k_p} = \left\{W_{p,\tau}^n : k_p \in \tau,\ \tau \subset \mathcal{K}_p, |\tau| = K\gamma, \forall n \in [N]\right\}$$

naturally corresponding to a normalized cache size

$$\frac{|Z_{k_p}|}{S} = \frac{\binom{K/L_c-1}{K\gamma-1}}{L_c \cdot \binom{K/L_c}{K\gamma}} = \frac{1}{L_c}\frac{K\gamma}{\frac{K}{L_c}} = \gamma. \tag{10}$$

### B. Delivery Phase

As mentioned above, the delivery of files to the users is based on the method of [17], which merges, in the same transmission, cache-aided and non-cache-aided users. The delivery is divided into $L_c$ sub-phases, where in sub-phase $q \in [L_c]$ the objective is to deliver to all users, the part of their requested file corresponding to partition (labeled by) $q$. Hence, users of group $\mathcal{K}_q$ act as the cache-aided users, while the remaining users, $[K] \setminus \mathcal{K}_q$, act as cache-less users.

We will focus on describing the delivery for one of the $L_c$ data parts (for part $q \in [L_c]$); for the other parts, corresponding to different $q$, we simply exchange the roles of the cache-aided and the cache-less users. We remind that the goal is to transmit simultaneously to $L + K\gamma$ users. To do so, we create an $L$ dimensional data vector, where one of its entries is the standard XOR (cf. [1])

$$X_\sigma = \bigoplus_{k\in\sigma} W_{\sigma\setminus\{k\}}^{r_k},\ \sigma \subseteq \mathcal{K}_q,\ |\sigma| = K\gamma+1 \tag{11}$$

intended for some $K\gamma+1$ ('cache-aided') users in $\mathcal{K}_q$, while the remaining $L-1$ entries of the data vector are $L-1$ uncoded subfiles intended for set $G_n \subseteq [K] \setminus \mathcal{K}_q$ of $L-1$ 'cache-less' users, where these $L-1$ uncoded subfiles are carefully picked

[4]Since memory sharing can be used as in [1].

[5]Naturally the subpacketization savings compared to other multi-antenna coded caching algorithms (see [4]–[7]), are even larger.

[6]We note that this DoF can be achieved while $K_n \le \frac{L-1}{\gamma}$.

to have the same index $\tau \subset \sigma$. This data vector is then ZF precoded, and the transmitted vector takes the form

$$\mathbf{x}_\sigma^\tau = \mathcal{H}_{\{k_q\}\cup G_n}^{-1} \begin{bmatrix} X_\sigma, \\ W_\tau^{r_{G_n(1)}}, \\ ..., \\ W_\tau^{r_{G_n(L)}} \end{bmatrix}^T. \tag{12}$$

Decoding at the cache-less users directly benefits from the ZF precoder, which delivers one stream to each of the $L-1$ cache-less users, and one stream (the XOR) to the cache-aided user $k_q$ who will subsequently 'cache-out' the interfering messages from the XOR to get its desired message $W_{\sigma\setminus\{k_q\}}^{r_{k_q}}$. On the other hand, any other 'cache-aided' receiver $k \in \tau$, will not benefit from precoding and will rather receive maximal interference in the form

$$\begin{aligned} y_\sigma^\tau(k\in\tau) &= \mathbf{h}_k^T \mathbf{x}_\sigma^\tau + w_k \\ &= \mathbf{h}_k^T \mathbf{h}_{G_n}^{-1} X_\sigma + \mathbf{h}_k^T \sum_{i\in G_n} \mathbf{h}_{G_n\setminus\{i\}\cup\{k_q\}}^{-1} W_\tau^{r_i} + w_k. \end{aligned}$$

Transmitted messages $W_\tau^{r_i}$ intended for the cache-less users have been picked to be completely known at all the cache-aided users in $\tau = \sigma \setminus \{k_q\}$, allowing each user in $\tau$ to cache-out these messages, with the additional use of their acquired CSIR[7]. This leaves each user in $\tau \subset \sigma$ with receiving only $X_\sigma$, from which they can naturally decode their desired message.

### C. Matching Algorithm

In the previous section we described the data vector that consists of a XOR $X_\sigma$ and $L-1$ uncoded subfiles each indexed with $\tau$. In this section we focus on how the XOR/subfile-index pairs are picked, so that the decoding process is successful at all $L + K\gamma$ users.

Focusing on a specific dataset part labeled by $q \in [L_c]$, the goal is to successfully communicate all possible XORs $X_\sigma,\ \sigma \subseteq \mathcal{K}_q,\ |\sigma| = 1+K\gamma$ (for all 'cache-aided' users in $\mathcal{K}_q$) and at the same time to communicate each subfile $W_\tau^{r_i},\ \forall i \in [K] \setminus \mathcal{K}_q,\ \forall \tau \subset \mathcal{K}_q,\ |\tau| = K\gamma$ for the remaining users.

We begin by forming a bipartite graph, where nodes on the right-hand-side (RHS) represent each one of the XORs, while each node on the left-hand-side (LHS) represents a set of $L-1$ subfiles with the same index $\tau$, but belonging to a different file (each intended for a different user).

The set of all nodes $\mathcal{L}_{p,\tau}$ on the LHS of the graph is

$$\left\{ \mathcal{L}_{p,\tau} :\ p \in \left[ \frac{K}{L+K\gamma} \right],\ \tau \subset \mathcal{K}_q,\ |\tau| = K\gamma \right\} \tag{13}$$

where $p$ designates a class of $L-1$ cache-less users (there are $\frac{1}{L-1}(K - \frac{K}{L_c}) = \frac{K}{L+K\gamma}$ such classes), while $\tau$ designates the subfile index.

On the other hand, the RHS of the graph consists of two types of nodes. The first set of nodes $\mathcal{R}_\sigma$ takes the form

$$\{\mathcal{R}_\sigma : \sigma \subseteq \mathcal{K}_q,\quad |\sigma| = K\gamma + 1\} \tag{14}$$

and each node corresponds to a XOR $X_\sigma$, while the second set of nodes $\mathcal{R}_{\emptyset,s}$ takes the form

$$\left\{ \mathcal{R}_{\emptyset,s} : s \in \left[ \frac{K\gamma}{1+K\gamma} \binom{K/L_c}{K\gamma} \right] \right\} \tag{15}$$

and each node corresponds to an empty message.

The problem at hand is to match each node $\mathcal{L}_{p,\tau}$ to a single and unique node $\mathcal{R}_\sigma$, which is equivalent to finding a *perfect matching* (cf. [18]). Each class-index pair $\mathcal{L}_{p,\tau}$ can share an edge with node $\mathcal{R}_\sigma$ iff $\tau \subset \sigma$ ( $\forall p \in \left[ \frac{K}{K\gamma+L} \right]$). Further, $\mathcal{L}_{p,\tau}$ can share an edge with any $\mathcal{R}_{\emptyset,s}$.

Thus, there are $\frac{K\gamma}{1+K\gamma}\binom{K/L_c}{K\gamma}$ possible edges from any LHS node $\mathcal{L}_{p,\tau}$, and these are the edges to any node of the second type of RHS nodes and to exactly $\frac{K}{L_c} - K\gamma$ nodes of the first type of RHS nodes[8]. Since each node has the same number of edges, then there exists a perfect matching (see [18]), while a perfect matching can be calculated through the low complexity algorithm of [19].

Calculating a perfect matching indicates which XOR $X_\sigma$ or which empty message will be transmitted with a class-index pair $(p, \tau)$. Thus, we transmit all XORs and their respective subfiles, while for pairs that are matched to an empty set we only transmit uncoded subfiles.

*Calculation of Delivery Time:* Focusing on the dataset partition labeled by $q$, we can observe that the transmission will end when all cache-less users (i.e., those in set $[K] \setminus \mathcal{K}_q$) receive all their requested subfiles. The fact that in each transmission we communicate a subfile to a set of $L-1$ users, implies that the number of transmissions is equal to the number of LHS nodes, which in turn implies a delay of

$$T = \frac{L_c \frac{K}{L+K\gamma}\binom{K/L_c}{K\gamma}}{L_c\binom{K/L_c}{K\gamma}} = \frac{K}{L+K\gamma}. \tag{16}$$

### D. Example

We assume a MISO BC setting where the transmitter has access to $L = 7$ antennas and $K = 18$ users each equipped with a cache of normalized size of $\gamma = \frac{1}{9}$. In this setting the subpacketization required by [1] is $S_1 = \binom{18}{3} = 153$, while this work requires a subpacketization of $S_L = 3\binom{6}{2} = 45$ and a CSIT cost of $C = L = 7$ training slots.

We begin with the placement phase, where users are divided into $L_c = \frac{L+K\gamma}{1+K\gamma} = 3$ groups i.e., $\mathcal{K}_1 = \{1,2,3,4,5,6\}$, $\mathcal{K}_2 = \{7,8,9,10,11,12\}$ and $\mathcal{K}_3 = \{13,14,15,16,17,18\}$. Further, each subfile is divided into $S = L_c\binom{K/L_c}{K\gamma}$ subfiles $W_{q,\tau}^n,\ q \in [L_c],\ \tau \subset \mathcal{K}_p,\ |\tau| = K\gamma$. Focusing on the first part of the dataset partition (i.e., focusing on the case of $q = 1$), for any file $n \in [N]$, the users cache as follows

$$\begin{aligned} Z_1 &= \{W_{1,12}^n, W_{1,13}^n, W_{1,14}^n, W_{1,15}^n, W_{1,16}^n\} \\ Z_2 &= \{W_{1,12}^n, W_{1,23}^n, W_{1,24}^n, W_{1,25}^n, W_{1,26}^n\} \\ Z_3 &= \{W_{1,13}^n, W_{1,23}^n, W_{1,34}^n, W_{1,35}^n, W_{1,36}^n\} \end{aligned}$$

[7]The fact that $L$ uplink and downlink training slots can provide for this (global) CSIR, is easy to see and it is discussed in [7].

[8]This happens because $(p,\tau) \leftrightarrow \sigma$ iff $\tau \subset \sigma$, thus for a specific $\tau$ we have $\sigma = \tau \cup \{k\}\ : k \in \mathcal{K}_q \setminus \tau$. This means that a $\tau$ can be matched to $|\mathcal{K}_q \setminus \tau| = \frac{K}{L_c} - K\gamma$ nodes.

$$Z_4 = \{W^n_{1,14}, W^n_{1,24}, W^n_{1,34}, W^n_{1,45}, W^n_{1,46}\}$$
$$Z_5 = \{W^n_{1,15}, W^n_{1,25}, W^n_{1,35}, W^n_{1,45}, W^n_{1,56}\}$$
$$Z_6 = \{W^n_{1,16}, W^n_{1,26}, W^n_{1,36}, W^n_{1,46}, W^n_{1,56}\}$$
$$Z_7 = ... = Z_{16} = \emptyset.$$

In order to determine the transmission pairs, we need to find a perfect matching over the formed bipartite graph.

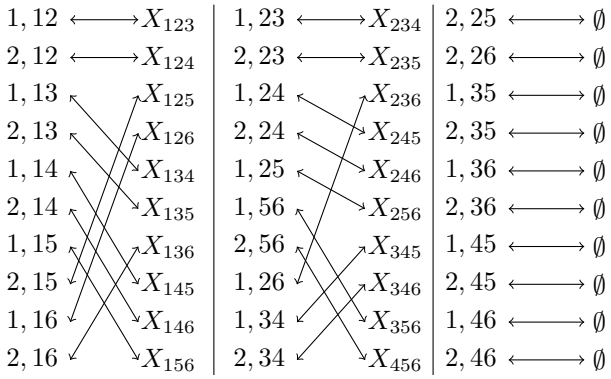


Fig. 1. The matching of the example of Sec. V-D split into three subgraphs.

*Delivery and Decoding:* The bipartite graph for part $q = 1$, and one possible matching, are presented in Fig. 1. As discussed above, a node on the LHS corresponds to a class of $L - 1 = 6$ cache-less users and the subfile each will receive, while nodes on the RHS represent the XORs, each destined to $K\gamma + 1$ users. The matching informs us of the set of $K\gamma + L$ users and the subfiles that need to be transmitted. For example, selecting the first XOR-subfile pair, we create the transmission vector

$$x^{12}_{123} = \mathcal{H}^{-1}_{\lambda} \begin{bmatrix} W^{r_1}_{1,23} \oplus W^{r_2}_{1,13} \oplus W^{r_3}_{1,12} \\ W^{r_7}_{1,12} \\ \vdots \\ W^{r_{12}}_{1,12} \end{bmatrix}$$

where $\lambda = \{3, 7, 8, 9, 10, 11, 12\}$. Decoding at users in $\lambda$ is direct since $\mathcal{H}^{-1}_{\lambda}$ separates the messages into $L$ parallel streams. From these users, user 3 will also then use its cache to extract its desired message from the XOR. The remaining users 1 and 2, receive a linear combination of $L$ elements. For example, user 1 will receive

$$y^{12}_{123}(1) = \mathbf{h}^T_1 \mathbf{h}^{-1}_{\lambda\setminus\{3\}} W^{r_1}_{1,23} \oplus W^{r_2}_{1,13} \oplus W^{r_3}_{1,12} + \\ + \mathbf{h}^T_1 \sum_{k=7}^{12} \mathbf{h}^{-1}_{\lambda\setminus\{k\}} W^{r_k}_{1,12} + w_1$$

where we can see that the contents of the last summation are cached at user 1, who can then remove them, and proceed to again use its cache to extract its desired information from the remaining XOR. A similar process takes place at user 2.

*Calculation of Time:* The bipartite graph of Fig. 1 (corresponding to part $p = 1$) implies 30 transmissions per part, which means that there will be a total of 90 transmissions, each of normalized duration $\frac{1}{S} = \frac{1}{45}$. Consequently the overall delivery time is $T = \frac{90}{45} = 2$.

## V. Concluding Remarks

In this work, we presented a new multi-antenna coded caching algorithm that has minimal feedback requirements ($C = L$ uplink and $C = L$ downlink training slots per-transmission), while also exponentially improving the required subpacketization compared to the single antenna case.

*Open Problems:* It is important to notice that while the feedback costs are significantly reduced, nonetheless the subpacketization reductions are much lower than those achieved in [8]. Thus, this current work can be viewed as a first step into exploiting the subpacketization-reduction benefits of multiple antennas, while requiring a small feedback cost.

It remains to be seen if these two opposing bottlenecks can further be ameliorated.

## References


[1] M. A. Maddah-Ali and U. Niesen, "Fundamental limits of caching," *IEEE Transactions on Information Theory*, 2014.

[2] 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. on Inf. Theory*, vol. 65, no. 1, pp. 647–663, Jan 2019.

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

[4] S. P. Shariatpanahi, S. A. Motahari, and B. H. Khalaj, "Multi-server Coded Caching," *IEEE Trans. on Inf. Theory*, 2016.

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

[6] E. Lampiris and P. Elia, "Achieving full multiplexing and unbounded caching gains with bounded feedback resources," in *IEEE Int. Symp. on Inf. Theory (ISIT)*, June 2018.

[7] ——, "Resolving a feedback bottleneck of multi-antenna Coded Caching," *arXiv preprint arXiv:1811.03935*, 2018.

[8] ——, "Adding transmitters dramatically boosts coded-caching gains for finite file sizes," *IEEE J. on Sel. Areas in Comm. (JSAC)*, 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*, 2016.

[10] Q. Yan, M. Cheng, X. Tang, and Q. Chen, "On the placement delivery array design for centralized coded caching scheme," *IEEE Transactions on Information Theory*, 2017.

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

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

[13] K. Shanmugam, A. M. Tulino, and A. G. Dimakis, "Coded Caching with linear subpacketization is possible using Ruzsa-Szemeredi graphs," in *IEEE Int. Symp. on Inf. Theory (ISIT)*, June 2017.

[14] P. Krishnan, "Coded caching via line graphs of bipartite graphs," in *IEEE Information Theory Workshop (ITW)*, Nov 2018.

[15] M. Cheng, J. Li, X. Tang, and R. Wei, "Linear coded caching scheme for centralized networks," *arXiv preprint arXiv:1810.06017*, 2018.

[16] L. Zheng and D. N. C. Tse, "Communication on the Grassmann manifold: A geometric approach to the noncoherent multiple-antenna channel," *IEEE Transactions on Information Theory*, 2002.

[17] E. Lampiris and P. Elia, "Full coded caching gains for cache-less users," in *IEEE Information Theory Workshop (ITW)*, Nov 2018.

[18] G. Agnarsson and R. Greenlaw, *Graph theory: Modeling, applications, and algorithms*. Pearson/Prentice Hall, 2007.

[19] A. Goel, M. Kapralov, and S. Khanna, "Perfect matchings in O($n\setminus \log n$) time in regular bipartite graphs," *SIAM Journal on Computing*, 2013.