Full Coded Caching Gains for Cache-less Users

Eleftherios Lampiris and Petros Elia

***Abstract*— The work identifies the performance limits of the multiple-input single-output broadcast channel where cache-aided users coincide with users that do not have caches. The main contribution is a new algorithm that employs perfect matchings on a bipartite graph to offer full multiplexing as well as full coded-caching gains to both cache-aided as well as cache-less users. This performance is shown to be within a factor of at most 3 from the optimal, under the assumption of linear one-shot schemes.**

An interesting outcome is the following: starting from a single-stream centralized coded caching setting with normalized cache size γ, then every addition of an extra transmit antenna allows for the addition of approximately $1/\gamma$ extra *cache-less* users, at no added delay costs. For example, starting from a single-stream coded caching setting with $\gamma = 1/100$, every addition of a transmit antenna, allows for serving approximately an additional 100 more cache-less users, without any increase in the overall delay. Finally the work reveals the role of multiple antennas in removing the penalties typically associated to cache-size unevenness, as it shows that the performance in the presence of both cache-less and cache-aided users, matches the optimal (under uncoded cache placement) performance of the corresponding symmetric case where the same cumulative cache volume is split evenly across all users.

I. Introduction

Coded caching is a technique — first introduced in [1] for the single-stream bottleneck broadcast channel (BC) — that exploits receiver-side caches in order to deliver cacheable content to many users at a time. This technique initially involved a setting where a single-antenna transmitter has access to a library of N files, and serves (via a single bottleneck link) K receivers, each having a cache of size equal to the size of M files. The process involved a novel cache placement method and a subsequent delivery phase during which each user simultaneously requests one library file, while the transmitter employs cache-dependent network coding to simultaneously deliver independent requested content to many users at a time.

In a normalized setting where the bottleneck link has capacity equal to 1 file per unit of time, the work in [1] showed that any set of K simultaneous requests can be served with *normalized* delay (worst-case completion time, guaranteeing the delivery of any set of requested files) which is at most $T = \frac{K(1-\gamma)}{1+K\gamma}$, where $\gamma \triangleq \frac{M}{N} \in [0,1]$ denotes the normalized cache size. This implied an ability to treat $K\gamma + 1$ cache-aided users at a time; a number that is often referred to as the cache-aided sum *degrees of freedom* (DoF) $d_\Sigma \triangleq \frac{K(1-\gamma)}{T}$, corresponding to a caching gain of $K\gamma$ additional served users due to caching.

Multi-antenna coded caching: Recently coded caching has been explored in the presence of multiple antennas/transmitters. In the context of a fully-connected multiple-input single-output (MISO) BC, multi-antenna (L antennas) techniques were combined with coded caching to reveal new insights such as that i) multiplexing and caching gains can be combined additively [2], [3] to yield a sum-DoF of $d_\Sigma = L + K\gamma$, ii) multiple antennas can dramatically reduce subpacketization, thus allowing for multiplicative DoF gains in the finite file-size regime [4], and iii) the feedback cost of combining the two gains is a function only of L (not of the caching gain) [5], as well as other insights (cf. [6]–[15], etc).

A. Coded caching in the presence of cacheless users

We here study the Degrees of Freedom (DoF) performance of the L-antenna K-user MISO BC[1], where K_c of the users are endowed with caches of normalized size $\gamma \in (0,1)$, while the rest K_n users have no cache.

As usual, each user simultaneously asks for a single – different – file, from a library of $N \geq K$ files. The received signal at each receiver k takes the form

$$y_k = \mathbf{h}_k^T \mathbf{x} + w_k, \quad \forall k \in \{1,2,...,K\} \triangleq [K]$$

where $\mathbf{h}_k$ denotes the $L \times 1$ channel from the transmitter to receiver k, $\mathbf{x}$ is the $L \times 1$ vector message transmitted from the L antenna array, and $w_k \sim \mathcal{N}(0,1)$ corresponds to the noise. We assume that each node has all necessary channel-state information, and that for a given signal-to-noise ratio (SNR), each link has capacity of the form $\log(SNR)+o(log(SNR))$.

The current scenario, where cache-aided users coexist with cache-less users, is of particular interest because, for example, some users may employ legacy devices that do not support cache-aided decoding, or because they may opt-out of dedicating their storage for caching (cf. [17]). Our aim is to design a pre-fetching and delivery algorithm that minimizes the normalized worst-case completion time $T_L(K_c, \gamma, K_n)$.

B. Notation

We will use $\mathcal{H}_\lambda^{-1}$ to denote the normalized inverse of the channel from the L antennas to the L users in the user set $\lambda \subset [K]$, $|\lambda| = L$. We will denote the requested file of user $k \in [K]$ with W_{d_k}, while $\mathcal{K}_c = \{1,2,...,K_c\}$ and $\mathcal{K}_n = \{K_c+1,...,K\}$ will denote the sets of cache-aided and cache-less users respectively. Symbol $\oplus$ denotes the bit-wise XOR operator and $\binom{n}{k}$ the binomial coefficient.

Eleftherios Lampiris and Petros Elia are with the Communication Systems Department at EURECOM, Sophia Antipolis, 06410, France (email: {lampiris, elia} @eurecom.fr. 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]We note that, while we here focus on the MISO BC, the results can be readily extended to the multiserver setting of [2] and the multiple transmitter setting where each transmitter stores some fraction of the content [3].

II. Main results

We start with a simple result that exemplifies — in the single stream case of $L = 1$ — the problem with having cache-aided users coexisting with cache-less users. We recall that for the single-stream case, $T_1(K_c, \gamma) = \frac{K_c(1-\gamma)}{1+K_c\gamma}$ corresponds to the optimal (under uncoded cache placement [18], [19]) delay needed to serve K_c cache-aided users.

Proposition 1. *In a single-stream BC with K_c cache-aided users with normalized cache size γ and with K_n additional cache-less users, the optimal delay takes the form*

$$T_1(K_c, \gamma, K_n) = T_1(K_c, \gamma) + K_n = \frac{K_c(1-\gamma)}{1+K_c\gamma} + K_n.$$

Sketch of proof: Due to lack of space, the proof will be presented in the journal version of this work. In brief, the proof is based on the approach in [18] that treats the case of $K_n = 0$, by translating the coded caching problem into an index coding problem and then creates an outer bound on $T_1(K_c, \gamma)$ by constructing large acyclic subgraphs of the side information graph for that index coding problem. Now, when $K_n > 0$, every graph node corresponding to a subfile requested by a cache-less user, will have no outgoing edges, and thus (this is a bit of a jump here) the new acyclic graph corresponding to $K_n > 0$ will consist of the acyclic graph corresponding to $K_n = 0$, plus all the aforementioned (directionally isolated) nodes that have a cumulative size equal to K_n files, which in turn translates to the above delay penalty of K_n. □

The above reveals that in the single stream case, every time a single cache-less user is added, there is a delay penalty of an entire unit of time, revealing that the two types of users can only be treated separately. If such separation were to be applied in the multi-antenna case, the performance would be

$$T_L(K_c, \gamma, K_n) = \frac{K_c(1-\gamma)}{L+K_c\gamma} + \frac{K_n}{L} \tag{1}$$

and the K_n cache-less users would experience no caching gain. We proceed with the main result. We will use $T_1 \triangleq T_1(K_c, \gamma)$.

Theorem 1. *In the MISO BC with $L > 1$ antennas, K_c cache-aided users, fractional cache size γ, and $K_n \geq (L-1)T_1$ cache-less users, the delivery time*

$$T = \frac{(L-1)T_1 + K_c(1-\gamma)}{K_c\gamma + L} + \frac{K_n - (L-1)T_1}{\min\{L, K_n - (L-1)T_1\}} \tag{2}$$

is achievable and within a factor of 2 from optimal, while if $K_n \leq (L-1)T_1(K_c, \gamma)$ then

$$T = \frac{K_n + K_c(1-\gamma)}{K_c\gamma + L} \tag{3}$$

is achievable and within a factor of 3 from optimal.

Proof. The achievability part of the proof can be found in Section III, while the outer bound and gap calculations can be found in the Appendix. □

Let us proceed to a few corollaries that explore some of the ramifications of the above theorem. Equation (1) helps us place the following corollary into context.

Corollary 1. *In the MISO BC with $K_n \leq (L-1)T_1(K_c, \gamma)$, all cache-aided and cache-less users can experience full multiplexing gain L as well as full caching gain $K_c\gamma$.*

Proof. The proof is direct from Equation (3). □

We proceed with another corollary which can be placed into context, by noting that in a cache-less system with L antennas and K_n *cache-less* users, adding one more antenna would allow (without added delay costs) the addition of only a diminishing number of $\frac{K_n}{L}$ extra cache-less users.

Corollary 2. *Starting from the basic single-stream BC with K_c cache-aided users of cache size γ, then adding an extra $L-1$ transmit antennas, allows for the addition of*

$$(L-1)T_1(K_c, \gamma) \approx \frac{L-1}{\gamma}$$

extra cache-less users, at no added delay costs.

Proof. This is direct from Theorem 1. □

We now show multiplicative DoF boosts from increasing L.

Corollary 3. *In the single-antenna (K_c, γ, K_n) BC with $K_n = (\tilde{L}-1)\frac{K_c(1-\gamma)}{1+K_c\gamma}$, going from 1 to $L \leq \tilde{L}$ antennas, reduces delay by L times.*

Proof. This is direct from Proposition 1 and Theorem 1. □

The following takes another point of view and explores the benefits of injecting cache-aided users into legacy (cache-less) MISO BC systems. To put the following corollary into context, we recall that in a cache-less L transmit-antenna MISO BC serving $K_n \geq L$ users, the optimal (normalized) delay is $\frac{K_n}{L}$.

Corollary 4. *In a MISO BC with $K_n \geq L$ cache-less users, introducing K_c additional cache-aided users with $\gamma \geq \frac{L}{K_n}$, incurs delay*

$$T_L(K_c, \gamma, K_n) \leq \frac{K_n}{L-1}$$

and thus we can add an infinite number of cache-aided users with a delay increase by a factor that is at most $\frac{L}{L-1}$.

Proof. This is direct from Theorem 1. □

Multiple antennas for 'balancing' cache-size unevenness: In the variety of works (cf. [20]–[23]) that explore the single-stream coded caching setting in the presence of uneven cache sizes, we see that having cache-size asymmetry induces delay penalties and that the preferred cache-size allocation is the uniform one. The following corollary addresses this issue, in the multi-antenna setting.

Corollary 5. *The (K_c, γ, K_n) MISO BC with $K_n \leq (L-1)\frac{K_c(1-\gamma)}{1+K_c\gamma}$ cache-less users, incurs the same achievable delay*

$$T(K_c, \gamma, K_n) = \frac{K_n + K_c(1-\gamma)}{L+K_c\gamma} = \frac{K(1-\gamma_{av})}{L+K\gamma_{av}} \tag{4}$$

as the optimal homogeneous K-user MISO BC with equal cache sizes $\gamma_{av} = \frac{K_c\gamma}{K}$ (same cumulative cache $K_c\gamma = K\gamma_{av}$).

Proof. This is direct from Theorem 1. □

III. Scheme Description

The challenge of the algorithm lies in the efficient combination of demands from cache-aided and cache-less users in the same transmission vector.

A. Cache placement

Before cache placement, each file $W_n,\ n \in [N]$, is divided into $K_c(1-\gamma)\binom{K_c}{K_c\gamma}$ subfiles as

$$W_n \to \{W^{\phi}_{n,\tau},\ \tau \subset \mathcal{K}_c,\ |\tau| = K_c\gamma,\ \phi \in \mathcal{K}_c \setminus \tau\}$$

and the cache Z_k of each user $k \in \mathcal{K}_c$ is filled according to

$$Z_k = \{W^{\phi}_{n,\tau} : k \in \tau, \forall \phi \in \mathcal{K}_c \setminus \tau,\ \forall n \in [N]\}. \quad (5)$$

This is identical to the original placement in [1], and the extra subpacketization (corresponding to ϕ) will facilitate the subsequent combinatorial problem of matching XORs with uncoded subfiles.

B. Delivery

We will first focus on the case of $K_n = (L-1)\frac{K_c(1-\gamma)}{K_c\gamma+1}$, where the delay $T = \frac{K_n+K_c(1-\gamma)}{K_c\gamma+L}$ can be achieved by simultaneously treating $K_c\gamma + L$ users. The extension to an arbitrary number K_n of cache-less users will be described later on.

Algorithm 1: Transmission Vectors

Set $T_1 = \frac{K_c(1-\gamma)}{1+K_c\gamma}$ (assume $T_1 \in \mathbb{N}$).

Group cacheless users:

$$g_1 = \{K_c+1, K_c+2, ..., K_c+L-1\}, ...,$$
$$g_{T_1} = \{K_c+(L-1)\cdot(T_1-1), ..., K\}.$$

for *all* $\tau \subset \mathcal{K}_c,\ |\tau| = K_c\gamma$ *(pick file index)* **do**
  for *all* $\phi \in \mathcal{K}_c \setminus \tau$ *(pick precoded user)* **do**
    Set $\chi = \tau \cup \{\phi\}$
    for *all* $t \in [T_1]$ *(pick cacheless group)* **do**
      Transmit:

$$\mathbf{x}^t_{\tau,\phi} = \mathcal{H}^{-1}_{\phi \cup g_t} \begin{bmatrix} \bigoplus_{k\in\chi} W^{\phi_k}_{d_k,\chi\setminus\{k\}} \\ W^{\phi}_{d_{g_t(1)},\tau} \\ \vdots \\ W^{\phi}_{d_{g_t(L-1)},\tau} \end{bmatrix},$$

      $\phi_k \in \chi \setminus \{k\},\ k \in \mathcal{K}_c$ are chosen arbitrarily so that all ϕ_k's are eventually picked for subfile $W^{\phi_k}_{d_k,\chi\setminus\{k\}}$.

a) Transmission: Delivery commences by identifying a group $\tau \subset \mathcal{K}_c$ of $K_c\gamma$ cache-aided users, that will not be assisted by precoding, and then an additional cache-aided user $\phi \in \mathcal{K}_c \setminus \tau$ that will be assisted by precoding. These are combined into a set $\chi = \tau \cup \{\phi\}$ of $K_c\gamma+1$ cache-aided users whose requested subfiles define a XOR $\bigoplus_{k\in\chi} W^{\phi_k}_{d_k,\chi\setminus\{k\}}$ with $K_c\gamma+1$ elements. Then, we pick one out of T_1 sets, comprized of $L-1$ cache-less users $g_t(1), \ldots, g_t(L-1)$, along with the vector of requested subfiles $W^{\phi}_{d_{g_t(1)},\tau}, \ldots, W^{\phi}_{d_{g_t(L-1)},\tau}$. Together with cache-aided user ϕ, these cache-less users form a 'precoding' set $\lambda = \{\phi\} \cup g_t$ which will define a precoding matrix $\mathcal{H}^{-1}_{\lambda}$, which will multiply the information vector $[\bigoplus_{k\in\chi} W^{\phi_k}_{d_k,\chi\setminus\{k\}},\ W^{\phi}_{d_{g_t(1)},\tau}, \ldots, W^{\phi}_{d_{g_t(L-1)},\tau}]^T$, thus forming the final transmitted vector $\mathbf{x}^t_{\tau,\phi}$.

b) Decoding: Due to the nature of $\mathcal{H}^{-1}_{\lambda}$, the $L-1$ cache-less users $g_t(1), \ldots, g_t(L-1)$ can instantly decode their message. User $\phi \in \mathcal{K}_c$ will be facilitated by precoding, and will thus only have to extract its message by caching out the rest of the subfiles in the XOR. This is immediate due to the nature of the XOR which comes directly from [1]. The remaining $K_c\gamma$ cache-aided users $\chi \setminus \phi = \tau$ (who are not assisted by precoding), will receive the XOR as well as the additional set of $L-1$ individual messages $W^{\phi}_{d_{g_t(1)},\tau}, \ldots, W^{\phi}_{d_{g_t(L-1)},\tau}$. Naturally all these users in τ can remove this latter set $W^{\phi}_{d_{g_t(1)},\tau}, \ldots, W^{\phi}_{d_{g_t(L-1)},\tau}$ (this is why all the subfiles for the currently treated cache-less users share the same index τ), and also naturally all the users in τ can detangle the XOR to get their own message.

c) Matching: As we have seen, the demands of the cache-aided users are treated by default by sending all $\chi = \tau \cup \{\phi\}$ transmissions (just like in [1]). These same transmissions must now serve all files $W^{\phi}_{d_{g_t(1)},\tau}, \ldots, W^{\phi}_{d_{g_t(L-1)},\tau}$ for all cache-less users, which means that we must guarantee that each $\tau \subset [K_c],\ |\tau| = K_c\gamma$ is matched to $\binom{K_c-K_c\gamma}{K_c\gamma+1-K_c\gamma} = K_c(1-\gamma)$ different transmission indices $\chi \subset [K_c],\ |\chi| = 1 + K_c\gamma$. This corresponds to a perfect matching problem over a bipartite graph[2], and we know from [24] that such matchings exist. Due to the complexity of finding such algorithms, which is generally high, we proposed here the matching in Algorithm 1, which asks for slightly higher subpacketization[3], but which has very low complexity.

C. Generalization of scheme and calculation of delay

We have seen that Algorithm 1 works for the case of $K_n = (L-1)T_1$ (recall $T_1 = T_1(K_c, \gamma)$), and treats $L + K_c\gamma$ users at a time (there is no data repetition), thus completing the delivery to all users with delay $T = \frac{K_n+K_c(1-\gamma)}{K_c\gamma+L}$.

For the case of $K_n > (L-1)T_1$, delivery is split in two parts. In the first part, we simply employ Algorithm 1 on the first $(L-1)T_1$ cache-less users while simultaneously completing the delivery to all K_c cache-aided users. This is done at a rate of $K_c\gamma+L$ users at a time. Then in the second part we treat the remaining $K_n-(L-1)T_1$ cache-less users via ZF-precoding, L users at a time. The above sum up to total delay

$$T = \frac{(L-1)T_1+K_c(1-\gamma)}{K_c\gamma+L} + \frac{K_n-(L-1)T_1}{\min\{L, K_n-(L-1)T_1\}}.$$

[2]We know that this is a matching problem as no two edges (τ) can be connected to the same vertices (χ), and we know that the matching is perfect as it covers the entire set of vertices (as all transmissions must carry files for cache-less users). We also know that the graph is a k-regular bipartite graph where $k = K_c(1-\gamma)$, since each τ must be included in a total of k transmissions.

[3]This subpacketization is higher than the one in the original work of [1] by a multiplicative factor of $K_c(1-\gamma)$ which impl

For the case of $K_n < (L-1)T_1(K_c,\gamma)$, we start with Algorithm 1, serving $K_c\gamma + L$ users at a time, until we 'run out' of files for cache-less users. At that point, having to treat only cache-aided users, we transition to a multi-antenna coded caching algorithm from [2]–[5] that will again treat $K_c\gamma+L$ users at a time. The overall process treats consistently $K_c\gamma + L$ users at a time, yielding the corresponding delay of $T = \frac{K_n+K_c(1-\gamma)}{K_c\gamma+L}$ as in (3). This concludes the proof of the achievability part for Theorem 1.

D. Example

We will consider the case where $L = 2$, $K_c = 5$, $\gamma = \frac{1}{5}$, $K_n = 2$, where the cache-aided users $\mathcal{K}_c = \{1,2,3,4,5\}$ respectively request files A, B, C, D, E, and the cache-less users $\mathcal{K}_n = \{6,7\}$ respectively request files F, G. First each file W_n is subpacketized into $\{W^{\phi}_{n,\tau},\ \tau \subset [K_c],\ |\tau| = K_c\gamma,\ \phi \in [K_c] \setminus \tau\}$, i.e., into $\{W^{\phi}_{n,1}, W^{\phi}_{n,2}, W^{\phi}_{n,3}, W^{\phi}_{n,4}, W^{\phi}_{n,5}\}$, $\forall \phi \in [K_c] \setminus \tau$. For example, file A is subpacketized into $A^2_1, A^3_1, A^4_1, A^5_1, A^1_2, A^3_2, A^4_2, A^5_2, A^1_3,\ A^2_3, A^4_3, A^5_3, A^1_4, A^2_4, A^3_4, A^5_4, A^1_5, A^2_5, A^3_5, A^4_5$.

As noted in (5), each cache Z_k is filled as $Z_k = \{W^{\phi}_{n,\tau} : k \in \tau, \forall \phi \in [K_c] \setminus \tau,\ \forall n \in [N]\}$ so for example, the first cache will contain

$$Z_1 = \left\{W^2_{n,1}, W^3_{n,1}, W^4_{n,1}, W^5_{n,1},\quad \forall n \in [N]\right\}.$$

a) Transmission and decoding for specific set of users: We will be treating $K_c\gamma + K_n = 3$ users at a time. Let us look in detail at the first case, where we treat cache-aided users $1, 2$ together with cache-less user 6. In this case, we transmit

$$\mathbf{x}^1_{1,2} = \mathcal{H}^{-1}_{26}\begin{bmatrix} A^1_2 B^2_1 \\ F^2_1 \end{bmatrix}. \tag{6}$$

For decoding, user 2 will receive — due to ZF precoding with the inverse $\mathcal{H}^{-1}_{26}$ to the channel to users $2, 6$ — only the XORed message $A^1_2 B^2_1 \triangleq A^1_2 \oplus B^2_1$, and will cache-out A^1_2 to decode the desired B^2_1. User 6 will receive, again due to precoding, only its respective desired message. User 1 will receive a linear combination of $A^1_2 \oplus B^2_1$ with F^2_1, and will cache-out B^2_1 and F^2_1 (note that '1' appears in both subfile indices) to decode the desired A^1_2.

b) Sequence of transmissions: We now proceed with the entire sequence of 40 transmissions. Given that each file is subpacketized into $K_c(1-\gamma)\binom{K_c}{K_c\gamma} = 5(1-\frac{1}{5})\binom{5}{1} = 20$ subpackets, the 40 transmissions will correspond to the desired delay of $T = \frac{K_n+K_c(1-\gamma)}{L+K_c\gamma} = 2$. The transmissions are:

$$\mathbf{x}^1_{1,2} = \mathcal{H}^{-1}_{26}\begin{bmatrix} A^1_2 B^2_1 \\ F^2_1 \end{bmatrix}, \qquad \mathbf{x}^2_{1,2} = \mathcal{H}^{-1}_{27}\begin{bmatrix} A^3_2 B^3_1 \\ G^2_1 \end{bmatrix}$$
$$\mathbf{x}^1_{1,3} = \mathcal{H}^{-1}_{36}\begin{bmatrix} A^1_3 C^3_1 \\ F^3_1 \end{bmatrix}, \qquad \mathbf{x}^2_{1,3} = \mathcal{H}^{-1}_{37}\begin{bmatrix} A^2_3 C^2_1 \\ G^3_1 \end{bmatrix}$$
$$\mathbf{x}^1_{1,4} = \mathcal{H}^{-1}_{46}\begin{bmatrix} A^1_4 D^4_1 \\ F^4_1 \end{bmatrix}, \qquad \mathbf{x}^2_{1,4} = \mathcal{H}^{-1}_{47}\begin{bmatrix} A^2_4 D^2_1 \\ G^4_1 \end{bmatrix}$$
$$\mathbf{x}^1_{1,5} = \mathcal{H}^{-1}_{56}\begin{bmatrix} A^1_5 E^5_1 \\ F^5_1 \end{bmatrix}, \qquad \mathbf{x}^2_{1,5} = \mathcal{H}^{-1}_{57}\begin{bmatrix} A^2_5 E^2_1 \\ G^5_1 \end{bmatrix}$$
$$\mathbf{x}^1_{2,1} = \mathcal{H}^{-1}_{16}\begin{bmatrix} A^4_2 B^4_1 \\ F^1_2 \end{bmatrix}, \qquad \mathbf{x}^2_{2,1} = \mathcal{H}^{-1}_{17}\begin{bmatrix} A^5_2 B^5_1 \\ G^1_2 \end{bmatrix}$$
$$\mathbf{x}^1_{2,3} = \mathcal{H}^{-1}_{36}\begin{bmatrix} B^2_3 C^3_2 \\ F^3_2 \end{bmatrix}, \qquad \mathbf{x}^2_{2,3} = \mathcal{H}^{-1}_{37}\begin{bmatrix} B^1_3 C^1_2 \\ G^3_2 \end{bmatrix}$$
$$\mathbf{x}^1_{2,4} = \mathcal{H}^{-1}_{46}\begin{bmatrix} B^2_4 D^4_2 \\ F^4_2 \end{bmatrix}, \qquad \mathbf{x}^2_{2,4} = \mathcal{H}^{-1}_{47}\begin{bmatrix} B^1_4 D^1_2 \\ G^4_2 \end{bmatrix}$$
$$\mathbf{x}^1_{2,5} = \mathcal{H}^{-1}_{56}\begin{bmatrix} B^2_5 E^5_2 \\ F^5_2 \end{bmatrix}, \qquad \mathbf{x}^2_{2,5} = \mathcal{H}^{-1}_{57}\begin{bmatrix} B^1_5 E^1_2 \\ G^5_2 \end{bmatrix}$$
$$\mathbf{x}^1_{3,1} = \mathcal{H}^{-1}_{16}\begin{bmatrix} A^4_3 C^4_1 \\ F^1_3 \end{bmatrix}, \qquad \mathbf{x}^2_{3,1} = \mathcal{H}^{-1}_{17}\begin{bmatrix} A^5_3 C^5_1 \\ G^1_3 \end{bmatrix}$$
$$\mathbf{x}^1_{3,2} = \mathcal{H}^{-1}_{26}\begin{bmatrix} B^4_3 C^4_2 \\ F^2_3 \end{bmatrix}, \qquad \mathbf{x}^2_{3,2} = \mathcal{H}^{-1}_{27}\begin{bmatrix} B^5_3 C^5_2 \\ G^2_3 \end{bmatrix}$$
$$\mathbf{x}^1_{3,4} = \mathcal{H}^{-1}_{46}\begin{bmatrix} C^3_4 D^4_3 \\ F^4_3 \end{bmatrix}, \qquad \mathbf{x}^2_{3,4} = \mathcal{H}^{-1}_{47}\begin{bmatrix} C^1_4 D^1_3 \\ G^4_3 \end{bmatrix}$$
$$\mathbf{x}^1_{3,5} = \mathcal{H}^{-1}_{56}\begin{bmatrix} C^3_5 E^5_3 \\ F^5_3 \end{bmatrix}, \qquad \mathbf{x}^2_{3,5} = \mathcal{H}^{-1}_{57}\begin{bmatrix} C^1_5 E^1_3 \\ G^5_3 \end{bmatrix}$$
$$\mathbf{x}^1_{4,1} = \mathcal{H}^{-1}_{16}\begin{bmatrix} A^3_4 D^3_1 \\ F^1_4 \end{bmatrix}, \qquad \mathbf{x}^2_{4,1} = \mathcal{H}^{-1}_{17}\begin{bmatrix} A^5_4 D^5_1 \\ G^1_4 \end{bmatrix}$$
$$\mathbf{x}^1_{4,2} = \mathcal{H}^{-1}_{26}\begin{bmatrix} B^3_4 D^3_2 \\ F^2_4 \end{bmatrix}, \qquad \mathbf{x}^2_{4,2} = \mathcal{H}^{-1}_{27}\begin{bmatrix} B^5_4 D^5_2 \\ G^2_4 \end{bmatrix}$$
$$\mathbf{x}^1_{4,3} = \mathcal{H}^{-1}_{36}\begin{bmatrix} C^2_4 D^2_3 \\ F^3_4 \end{bmatrix}, \qquad \mathbf{x}^2_{4,3} = \mathcal{H}^{-1}_{37}\begin{bmatrix} C^5_4 D^5_3 \\ G^3_4 \end{bmatrix}$$
$$\mathbf{x}^1_{4,5} = \mathcal{H}^{-1}_{56}\begin{bmatrix} D^4_5 E^5_4 \\ F^5_4 \end{bmatrix}, \qquad \mathbf{x}^2_{4,5} = \mathcal{H}^{-1}_{57}\begin{bmatrix} D^1_5 E^1_4 \\ G^5_4 \end{bmatrix}$$
$$\mathbf{x}^1_{5,1} = \mathcal{H}^{-1}_{16}\begin{bmatrix} A^3_5 E^3_1 \\ F^1_5 \end{bmatrix}, \qquad \mathbf{x}^2_{5,1} = \mathcal{H}^{-1}_{17}\begin{bmatrix} A^4_5 E^4_1 \\ G^1_5 \end{bmatrix}$$
$$\mathbf{x}^1_{5,2} = \mathcal{H}^{-1}_{26}\begin{bmatrix} B^3_5 E^3_2 \\ F^2_5 \end{bmatrix}, \qquad \mathbf{x}^2_{5,2} = \mathcal{H}^{-1}_{27}\begin{bmatrix} B^4_5 E^4_2 \\ G^2_5 \end{bmatrix}$$
$$\mathbf{x}^1_{5,3} = \mathcal{H}^{-1}_{36}\begin{bmatrix} C^2_5 E^2_3 \\ F^3_5 \end{bmatrix}, \qquad \mathbf{x}^2_{5,3} = \mathcal{H}^{-1}_{37}\begin{bmatrix} C^4_5 E^4_3 \\ G^3_5 \end{bmatrix}$$
$$\mathbf{x}^1_{5,4} = \mathcal{H}^{-1}_{46}\begin{bmatrix} D^2_5 E^2_4 \\ F^4_5 \end{bmatrix}, \qquad \mathbf{x}^2_{5,4} = \mathcal{H}^{-1}_{47}\begin{bmatrix} D^3_5 E^3_4 \\ G^4_5 \end{bmatrix}.$$

The 40 slots, each of duration $t_s = \left(K_c(1-\gamma)\binom{K_c}{K_c\gamma}\right)^{-1} = \frac{1}{20}$, imply a delay $T = 2$, which is also the same delay that would be needed in the symmetric case where the $K = 7$ users would have an identical $\gamma_{av} = \frac{1}{7}$ (same cumulative $K\gamma_{av} = 1$).

IV. Conclusion and Final Remarks

An interesting outcome (Corollary 3) is the fact that despite having abundant side information at a sizable number K_c of receivers, going from 1 to L antennas gives an L-fold DoF boost. This comes in obvious contrast to the cache-aided multiple antenna setting with only cache-aided users [2], [3], [5], where adding antennas increases additively and not multiplicatively the DoF. Another interesting outcome is the antenna-aided amelioration of cache-size asymmetries. This can be important in practical scenarios where γ is expected to be small, which would then allow cache-aided users to boost the DoF performance of a large number ($\approx 1/\gamma$) of cache-less users.

Design Intuition: The scheme is based on the realization that *not all* served users required caches, which allowed substituting $L-1$ cache-aided users per transmission with

cache-less users, thus allowing these caches to be utilized in later transmissions to boost performance.

V. Appendix: Converse and gap to optimal

Let us first consider the gap to optimal for the case of $K_n \geq (L-1)T_1$, where recall that T_1 is short for $T_1(K_c,\gamma)$. We have seen that when $K_n = \alpha(L-1)T_1 \geq (L-1)T_1$ (i.e., when $\alpha \geq 1$), the achievable delay in (2) took the form

$$T = \frac{(L-1)T_1+K_c(1-\gamma)}{K_c\gamma+L} + \frac{K_n-(L-1)T_1}{L} \quad (7)$$

$$= \frac{(L-1)T_1+K_c(1-\gamma)}{L+K\gamma} + (\alpha-1)\frac{L-1}{L}T_1 \quad (8)$$

$$= \frac{T_1}{L}(\alpha L-\alpha+1). \quad (9)$$

For a lower bound on the minimum possible delay, we use

$$T = \frac{\min\{K_n,L\}}{L} = \min\{1, \frac{\alpha(L-1)T_1}{L}\} \quad (10)$$

corresponding to the optimal delay required to satisfy only the cache-less users. A quick calculation of the ratio between (7) and (10), bounds the gap as

$$G = \frac{\alpha L-\alpha+1}{\alpha L-\alpha} = 1+\frac{1}{\alpha(L-1)} \leq 2. \quad (11)$$

When $K_n < L$, then $\alpha(L-1)T_1 < L$, which again gives

$$G = \frac{T_1}{L}(\alpha L-\alpha+1) < \frac{T_1(\alpha L-\alpha-1)}{\alpha(L-1)T_1} \leq 2. \quad (12)$$

For the case of $K_n = \alpha(L-1)T_1, \ \ \alpha \leq 1$, the lower bound takes the form

$$T \geq \max\left\{\frac{\min\{L,K_n\}}{L}, \ \frac{1}{2}\frac{K_c(1-\gamma)}{K_c\gamma+L}\right\} \quad (13)$$

where the first term corresponds to the optimal performance of an 'easier' system where all the cache-aided users are removed, and where the second term corresponds to an easier system where all cache-less users are removed, and where — for this latter type of system, we know from [3] that treating $K_c\gamma+L$ users at a time is at most a factor of 2 from optimal, under the assumptions of linear and one-shot schemes. Combining (13) with the achievable $T = \frac{K_n+K_c(1-\gamma)}{K_c\gamma+L}$ from (3), yields a gap of

$$G = \frac{\frac{K_n+K_c(1-\gamma)}{K_c\gamma+L}}{\max\left\{\frac{K_n}{L}, \frac{1}{2}\frac{K_c(1-\gamma)}{K_c\gamma+L}\right\}}.$$

To bound this gap, note that if $\frac{K_n}{L} > \frac{1}{2}\frac{K_c(1-\gamma)}{K_c\gamma+L}$ then we know from before that

$$G = \frac{\frac{K_n+K_c(1-\gamma)}{K_c\gamma+L}}{\frac{K_n}{L}} = \frac{L}{K_c\gamma} + \frac{\frac{K_c(1-\gamma)}{K_c\gamma+L}}{\frac{K_n}{L}} \leq 1+2,$$

where we used that $\frac{K_n}{L} > \frac{1}{2}\frac{K_c(1-\gamma)}{K_c\gamma+L}$.

Similarly when $\frac{K_n}{L} < \frac{1}{2}\frac{K_c(1-\gamma)}{K_c\gamma+L}$, the gap is bounded as

$$G = \frac{\frac{K_n+K_c(1-\gamma)}{K_c\gamma+L}}{\frac{1}{2}\frac{K_c(1-\gamma)}{K_c\gamma+L}} = \frac{\frac{K_n}{K_c\gamma+L}}{\frac{1}{2}\frac{K_c(1-\gamma)}{K_c\gamma+L}} + 2 \leq 3$$

where the last step considers that $\frac{K_n}{L} < \frac{1}{2}\frac{K_c(1-\gamma)}{K_c\gamma+L}$.

This concludes the proof of Theorem 1. □

References

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

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

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

[4] E. Lampiris and P. Elia, "Adding transmitters dramatically boosts Coded-Caching gains for finite file sizes," *to appear in Journal of Selected Areas in Communications (JSAC)*, 2018.

[5] E. Lampiris and P. Elia, "Achieving full multiplexing and unbounded caching gains with bounded feedback resources," *to appear in International Symposium of Information Theory (ISIT)*, 2018.

[6] M. Bayat, R. K. Mungara, and G. Caire, "Achieving spatial scalability for coded caching over wireless networks," *arXiv preprint arXiv:1803.05702*, 2018.

[7] K.-H. Ngo, S. Yang, and M. Kobayashi, "Scalable content delivery with coded caching in multi-antenna fading channels," *IEEE Transactions on Wireless Communications*, 2018.

[8] J. Zhang and P. Elia, "Fundamental limits of cache-aided wireless BC: Interplay of Coded-Caching and CSIT feedback," *IEEE Transactions on Information Theory*, 2017.

[9] X. Yi and G. Caire, "Topological coded caching," in *IEEE International Symposium on Information Theory (ISIT)*, 2016.

[10] J. Zhang and P. Elia, "Wireless Coded Caching: A topological perspective," in *IEEE Int. Symposium on Inf. Theory (ISIT)*, 2017.

[11] A. Sengupta, R. Tandon, and O. Simeone, "Cache aided wireless networks: Tradeoffs between storage and latency," in *Proc. of the Conference on Information Science and Systems (CISS)*, 2016.

[12] Y. Cao, M. Tao, F. Xu, and K. Liu, "Fundamental storage-latency tradeoff in cache-aided mimo interference networks," *IEEE Transactions on Wireless Communications*, 2017.

[13] J. S. P. Roig, D. Gündüz, and F. Tosato, "Interference networks with caches at both ends," in *IEEE International Conference on Communications (ICC)*, 2017.

[14] J. Zhang, F. Engelmann, and P. Elia, "Coded caching for reducing CSIT-feedback in wireless communications," in *Proc. Allerton Conference*, 2015.

[15] L. Tang and A. Ramamoorthy, "Coded caching for networks with the resolvability property," in *IEEE International Symposium on Information Theory (ISIT)*, 2015.

[16] S. S. Bidokhti, M. Wigger, and R. Timo, "Erasure broadcast networks with receiver caching," in *IEEE International Symposium on Information Theory (ISIT)*, 2016.

[17] G. Paschos, E. Bastug, I. Land, G. Caire, and M. Debbah, "Wireless caching: Technical misconceptions and business barriers," *IEEE Communications Magazine*, 2016.

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

[19] Q. Yu, M. A. Maddah-Ali, and A. S. Avestimehr, "The exact rate-memory tradeoff for caching with uncoded prefetching," *IEEE Transactions on Information Theory*, February 2017.

[20] S. Wang, W. Li, X. Tian, and H. Liu, "Fundamental limits of heterogenous cache," *CoRR, vol. abs/1504.01123*, 2015.

[21] M. M. Amiri, Q. Yang, and D. Gündüz, "Decentralized caching and coded delivery with distinct cache capacities," *IEEE Transactions on Communications*, 2017.

[22] A. Sengupta, R. Tandon, and T. C. Clanc, "Layered caching for heterogeneous storage," in *IEEE 50th Asilomar Conference on Signals, Systems and Computers*, 2016.

[23] A. M. Ibrahim, A. A. Zewail, and A. Yener, "Centralized coded caching with heterogeneous cache sizes," in *Wireless Communications and Networking Conference (WCNC)*, IEEE, 2017.

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