Adding Transmitters Allows Unbounded Coded-Caching Gains with Bounded File Sizes

Eleftherios Lampiris
EURECOM
06410 Biot, France
Email: lampiris@eurecom.fr

Petros Elia
EURECOM
06410 Biot, France
Email: elia@eurecom.fr

***Abstract*— In the context of coded caching in the K-user BC, our work reveals the surprising fact that having multiple (L) transmitting antennas, dramatically ameliorates the long-standing subpacketization bottleneck of coded caching by reducing the required subpacketization to approximately its Lth root, thus boosting the actual DoF by a *multiplicative* factor of up to L. In asymptotic terms, this reveals that as long as L scales with the theoretical caching gain, then the full cumulative (multiplexing + full caching) gains are achieved with constant subpacketization. This is the first time, in any known setting, that unbounded caching gains appear under finite file-size constraints. The achieved caching gains here are up to L times higher than any caching gains previously experienced in any single- or multi-antenna fully-connected setting, thus offering a multiplicative mitigation to a subpacketization problem that was previously known to hard-bound caching gains to small constants.**

The proposed scheme is practical and it works for all values of K, L and all cache sizes. The scheme's gains show in practice: e.g. for $K = 100$, when $L = 1$ the theoretical caching gain of $G = 10$, under the original coded caching algorithm, would have needed subpacketization $S_1 = \binom{K}{G} = \binom{100}{10} > 10^{13}$, while if extra transmitting antennas were added, the subpacketization was previously known to match or exceed S_1. Now for $L = 5$, our scheme offers the theoretical (unconstrained) cumulative DoF $d_L = L + G = 5 + 10 = 15$, with subpacketization $S_L = \binom{K/L}{G/L} = \binom{100/5}{10/5} = 190$. The scheme's performance, given subpacketization $S_L = \binom{K/L}{G/L}$, is within a factor of 2 from the optimal linear sum-DoF. The gains stemming from this work come by a virtual decomposition of the fully connected cache-aided channel into parallel ones, which significantly reduces the required subpacketization

I. Introduction

Coded caching is a communication method invented in [1] that exploits receiver-side caches in broadcast-type communications, to achieve substantial throughput gains by delivering independent content to many users at a time.

Specifically the work in [1] considered the single-stream broadcast channel (BC) scenario where a single-antenna transmitter has access to a library of N files, and serves K receivers, each having a cache of size equal to the size of M files. In a normalized setting where the link has capacity 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) which is at most $T = K(1-\gamma)/(1+K\gamma)$ where $\gamma \triangleq M/N$ denotes the normalized cache size. This was a major breakthrough as it implied a scaling sum-DoF of

$$d_1(\gamma) = K(1-\gamma)/T = 1 + K\gamma$$

users served at a time. Given that in the absence of caching, only one user could be served at a time (because $d_1(\gamma = 0) = 1$), the above implied a (theoretical) caching gain of

$$G = d_1(\gamma) - d_1(\gamma = 0) = K\gamma$$

corresponding to the number of extra users that could be served at a time, additionally, as a consequence of introducing caching.

This gain — which is known to be close to the information theoretic optimal [1] — was shown to persist or approximately persist, under a variety of settings that include uneven topologies [2]–[4], a variety of channels such as erasure channels [5], MIMO broadcast channels with fading [6], a variety of networks such as heterogeneous networks [7], and in other settings as well.

A. Subpacketization bottleneck of coded caching

While though in theory, this caching gain $G = K\gamma$ increased indefinitely with increasing K, in practice the gain remained — under most realistic assumptions — hard-bounded by small constants, due to the fact that the underlying coded caching algorithm required the splitting of finite-length files into an exponential number of subpackets. For the algorithm in [1] in the original single-stream scenario, the near-optimal (and under some basic assumptions, optimal [9], [10]) gain of $G = K\gamma$, was achieved only if each file was segmented at least into a total of

$$S_1 = \binom{K}{K\gamma} \tag{1}$$

subpackets. As a result, having a certain maximum-allowable subpacketization of S_{max}, implied that one could only encode over a maximum of

$$\bar{K} = \arg\max_{K^o \le K} \left\{ \binom{K^o}{K^o\gamma} \le S_{max} \right\} \tag{2}$$

users, which in turn implied a substantially reduced *effective caching gain* $\bar{G}_1$ of the form

$$\bar{G}_1 = \bar{K}\gamma. \tag{3}$$

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

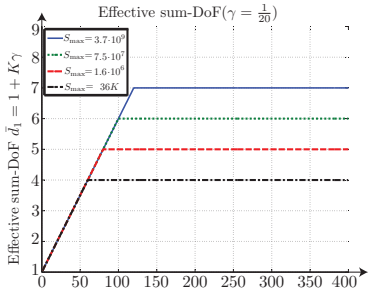

Fig. 1: Maximum effective DoF $\bar{d}_1$ achieved by the original centralized algorithm (single antenna, $\gamma = 1/20$) in the presence of different subpacketization constraints S_{max}. The gain is hard-bounded irrespective of K.

Given that

$$\binom{\bar{K}}{\bar{K}\gamma} \in \left[\left(\frac{1}{\gamma}\right)^{\bar{K}\gamma}, \left(\frac{e}{\gamma}\right)^{\bar{K}\gamma}\right] = \left[\left(\frac{1}{\gamma}\right)^{\bar{G}_1}, \left(\frac{e}{\gamma}\right)^{\bar{G}_1}\right] \tag{4}$$

this effective gain $\bar{G}_1$ was bounded as

$$\frac{\log S_{max}}{1+\log\frac{1}{\gamma}} \leq \bar{G}_1 \leq \frac{\log S_{max}}{\log\frac{1}{\gamma}}, \qquad \bar{G}_1 \leq G \tag{5}$$

(log is the natural logarithm) which succinctly reveals that the effective caching gain $\bar{G}_1$ (and the corresponding *effective sum-DoF* $\bar{d}_1 \triangleq 1+\bar{G}$) is placed under constant pressure from the generally small values[1] of γ and of S_{max}. This is reflected in Figure 1. Similar conclusions were highlighted in [12].

B. Coded caching with multiple transmitters

At the same time, different works (cf. [13], [14] as well as [6], [15]–[18] and others) aimed at complementing such caching gains, with additional multiplexing gains that can appear when there are several transmitters. One pioneering work in this direction was found in [13] which considered a setting with $L \leq K$ transmitters/servers communicating (in the fully-connected BC context of a so-called 'linear network' that can translate readily to a K-user wireless MISO BC with L antennas) to K single-antenna cache-aided receivers, which provided a scheme that achieved a theoretical sum-DoF of

$$d_L(\gamma) = L + K\gamma$$

corresponding to a MIMO multiplexing gain of L (users served, per second per hertz) and an additional theoretical caching gain of again $G = K\gamma$ (extra users served at a time, due to caching). This theoretical caching gain though was again restricted to an effective caching gain that was less than the effective gain $\bar{G}_1$ achieved in the single antenna case, because of a further increased subpacketization which now took the form $S = \binom{K}{K\gamma}\binom{K-K\gamma-1}{L-1}$.

While the subpacketization-constrained (effective) gains may have been reduced, this work in [13] nicely showed that multiplexing and caching gains are in theory additive.

Similar conclusions were drawn in [14] where — in the context of a cache-aided interference scenario where K_T transmitters with normalized cache size γ_T (each transmitter could only store a fraction γ_T of the entire N-file library), communicated to K receivers with normalized cache size γ — the work provided a scheme that achieved a sum-DoF of $\frac{K(1-\gamma)}{T} = K_T\gamma_T + K\gamma$ but with employed subpacketization

$$S = \binom{K}{K\gamma}\binom{K_T}{K_T\gamma_T}\binom{K-K\gamma-1}{L-1}. \tag{6}$$

In both cases [13], [14], adding extra dimensions on the transmitter side, maintained the theoretical caching gains, added extra multiplexing gains, but maintained high subpacketization levels with generally reduced actual caching gains.

To the best of our knowledge, under the generous assumptions that $S_{max} \leq 10^5$, $\gamma \leq 1/50$ and $K \leq 10^5$, currently there exists no method *in any known single-antenna or multi-antenna* fully connected setting, that allows for the introduction of more than $\bar{G} = 5$ additional users (per second per hertz, i.e., served at a time) due to caching[2].

C. Notation

For clarity, we recall the following notation: $d_1(\gamma) = 1+K\gamma$ (theoretical DoF for $L = 1$), $d_L(\gamma) = L + K\gamma$ (Theoretical DoF for multiple antennas), $d_L(\gamma = 0) = L$ (Multiplexing gain), $G = d_1(\gamma) - d_1(\gamma = 0) = d_L(\gamma) - d_L(\gamma = 0) = K\gamma$ (Theoretical caching gain)[3], $S_1 = \binom{K}{K\gamma}$ (Subpacketization needed for theoretical G for $L = 1$), S_{max} (Maximum allowable subpacketization), S_L (Subpacket. needed for theoretical G for multiple antennas), $\bar{d}_1(\gamma)$ (Effective (subpacketization constrained) DoF for $L = 1$), $\bar{G}_1 = \bar{d}_1(\gamma) - 1$ (Effective caching gain for $L = 1$), $\bar{d}_L(\gamma)$ (Effective DoF for multiple antennas), $\bar{G}_L = \bar{d}_L(\gamma) - L$ (Effective caching gain for multiple antennas). In the above, $\bar{d}_L(\gamma = 0) = d_L(\gamma = 0) = L$ is the multiplexing gain, and $\bar{G}_L$ is the effective caching gain describing the actual number of additional users that can be served at a time as a result of introducing caching, under a subpacketization constraint. Finally the effective DoF $\bar{d}_L(\gamma) = L + \bar{G}_L$ describes the actual (total) number of users that can be served at a time, under a subpacketization constraint.

[1] As argued in [11], in wireless cellular settings γ can be very small, which — for a given target caching gain — implies the need to code over many users, which in turn increases subpacketization. Compounding on this problem, there is a variety of factors that restrict the maximum allowable subpacketization level S_{max}. One such parameter is the file size; for example, movies are expected to have size that is close to or less than 1 Gigabyte. Additionally, in applications like video streaming, a video file it self may be broken down into smaller independent parts (on which subpacketization will take place separately), in order to avoid the delay that comes from the asynchronous nature of decoding XORs in coded caching. Such restricted file sizes may be in the order of just a few tens of Megabytes. Another parameter that restricts S_{max} is the minimum packet size; the atomic unit of storage is not a bit but a sector (newer 'Advanced Format' hard drives use 4096-byte sectors and force zero-padding on the remaining unused sector), and similarly the atomic communication block is the packet, which must maintain a certain minimum size in order to avoid communication delay overheads.

[2] This corresponds to Construction 6 in [19] ($a = b = 2, \lambda = 100$), and it requires approximately 20000 users.

[3] The choice here to measure the caching gain as the DoF difference $G = d_1(\gamma) - d_1(\gamma = 0) = d_L(\gamma) - d_L(\gamma = 0) = K\gamma$ rather than the DoF ratio, comes from the fact that in theory, the two gains (multiplexing and caching gains) appear to aggregate in an additive manner (this is noted also in [14]).

Furthermore we will use $[K] \triangleq \{1, 2, \cdots, K\}$. If $\mathcal{A}$ is a set, then $|\mathcal{A}|$ will denote its cardinality. For sets $\mathcal{A}$ and $\mathcal{B}$, then $\mathcal{A}\backslash\mathcal{B}$ denotes the difference set. The expressions $\alpha|\beta$ (resp. $\alpha \nmid \beta$) denote that integer α divides (resp. does not divide) integer β. Furthermore if $\mathcal{A} \subset [K]$ is a subset of users, then we will use $\mathbf{H}^{\mathcal{A}}$ to denote the overall channel from the L-antenna transmitter to the users in $\mathcal{A}$. Logarithms are of base e. In a small abuse of notation, we will sometimes denote data sets the same way we denote the complex numbers (or vectors) that carry that same data.

II. System and channel model

We consider the K-user multiple-input single-output (MISO) broadcast channel where an L-antenna transmitter communicates to K single-antenna receiving users. The transmitter has access to a library of N distinct files $W_1, W_2, \ldots, W_N$, each of size $|W_n| = f$ bits. Each user $k \in \{1, 2, \ldots, K\}$ has a cache Z_k, of size $|Z_k| = Mf$ bits, where naturally $M \leq N$. The delivery phase commences when each user k requests from the transmitter, any *one* file $W_{R_k} \in \{W_n\}_{n=1}^N$, out of the N library files. Upon notification of the users' requests, the transmitter aims to deliver the requested files, and do so with reduced duration T. During this delivery phase, for each transmission, the received signals at each user k, will be modeled as $y_k = \boldsymbol{h}_k^T \boldsymbol{x} + w_k, \;\; k = 1, \ldots, K$, where $\boldsymbol{x} \in \mathbb{C}^{L\times 1}$ denotes the transmitted vector satisfying a power constraint $\mathbb{E}(||\boldsymbol{x}||^2) \leq P$, where $\boldsymbol{h}_k \in \mathbb{C}^{L\times 1}$ denotes the channel of user k in the form of the random vector of fading coefficients that can change in time and space, and where w_k represents unit-power AWGN noise at receiver k. We will assume that P is high (high SNR), we will assume perfect channel state information throughout the (active) nodes as in [13], [14], and we will assume that the fading process is statistically symmetric across users.

As in [1], T is the number of time slots, per file served per user, needed to complete the delivery process, *for any request*. The wireless link capabilities, and the time scale, are normalized such that one time slot corresponds to the optimal amount of time it would take to communicate a single file to a single receiver, had there been no caching and no interference.

As in [1], we will first consider the case where $\gamma = \frac{M}{N} = \{1, 2, \cdots, K\}\frac{1}{K}$, while for non integer $K\gamma$, we will simply consider the result corresponding to $\lfloor K\gamma \rfloor$.

III. Description of the scheme

We will present the scheme for all K, γ, L, focusing on the aforementioned integer case where $L|K\gamma$ and $L|K$.

a) Grouping: We first split the K users $k = 1, 2, \ldots, K$ into $K' \triangleq \frac{K}{L}$ disjoint groups

$$\mathcal{G}_g = \{\ell K' + g, \;\; \ell = 0, 1, \ldots, L-1\}, \;\; \text{for } g = 1, 2, \ldots, K'$$

of $|\mathcal{G}_g| = L$ users per group. Our aim is to apply the algorithm of [1] to serve $K'\gamma + 1$ groups at a time, essentially treating each group as a single user. Toward this, let

$$\mathcal{T} = \{\tau \subset [K'] \;:\; |\tau| = K'\gamma\}$$

be the set of

$$|\mathcal{T}| = \binom{K'}{K'\gamma} \tag{7}$$

subsets in $[K']$, each of size $|\tau| = K'\gamma$, and let

$$\mathcal{X} = \{\chi \subseteq [K'] \;:\; |\chi| = K'\gamma + 1\}$$

be the set of $|\mathcal{X}| = \binom{K'}{K'\gamma+1}$ subsets of size $|\chi| = K'\gamma + 1$.

b) Subpacketization and caching: We first split each file W_n into $|\mathcal{T}|$ subfiles $\{W_n^\tau\}_{\tau\subset\mathcal{T}}$, and then we assign each user $k \in \mathcal{G}_g$ the cache

$$Z_k = Z_{\mathcal{G}_g} = \{W_n^\tau \;:\; \forall \tau \ni g\}_{n=1}^N \tag{8}$$

so that all users of the same group have an identical cache.

c) Transmission: After notification of requests — where each receiver k requires file W_{R_k}, $R_k \in [N]$ — the delivery consists of a sequential transmission $\{\boldsymbol{x}_\chi\}_{\chi\subseteq\mathcal{X}}$ where each transmission takes the form

$$\boldsymbol{x}_\chi = \sum_{g\in\chi} \sum_{k\in\mathcal{G}_g} W_{R_k}^{\chi\backslash g} \boldsymbol{v}^{\mathcal{G}_g\backslash k} \tag{9}$$

and where $\boldsymbol{v}^{\mathcal{G}_g\backslash k}$ is an $L\times 1$ precoding vector that is designed to belong in the null space of the channel $\mathbf{H}^{\mathcal{G}_g\backslash k}$ between the L-antenna transmitter and the $L-1$ receivers in group $\mathcal{G}_g$ excluding receiver $k \in \mathcal{G}_g$.

d) Decoding — 'Caching-out' out-of-group messages: The corresponding received signal at user $k \in \mathcal{G}_g$ is then

$$\boldsymbol{y}_{k,\chi} = \boldsymbol{h}_k^T \boldsymbol{x}_\chi + \boldsymbol{w}_{k,\chi} \tag{10}$$

and each such user $k \in \mathcal{G}_g$ can employ its cache to immediately remove all the files that are jointly undesired by its own group $\mathcal{G}_g$, i.e., receiver $k \in \mathcal{G}_g$ can remove $\sum_{g'\in\chi\backslash g} \sum_{j\in\mathcal{G}_{g'}} W_{R_j}^{\chi\backslash g'} \boldsymbol{v}^{\mathcal{G}_{g'}\backslash j}$ because $g' \neq g \in \chi$, i.e., because the cache of receiver k includes all files $W_{R_j}^{\chi\backslash g'}$ in the above summation. This allows receiver k to remove all files that are not of interest to its group $\mathcal{G}_g$, and thus to get

$$\boldsymbol{y}_{k,\chi}^{'} = \boldsymbol{h}_k^T \Big(\sum_{j\in\mathcal{G}_g} W_{R_j}^{\chi\backslash g} \boldsymbol{v}^{\mathcal{G}_g\backslash j} \Big) + \boldsymbol{w}_{k,\chi}. \tag{11}$$

e) Nulling-out intra-group messages — completion of decoding: The interference for receiver k now could only come from the files of the $L-1$ other users of its own group $\mathcal{G}_g$. This interference is averted directly by the ZF precoders (or any other DoF optimal precoder), and receiver k can get the desired $W_{R_k}^{\chi\backslash g}$.

This is done instantaneously for all users $k \in \mathcal{G}_g$, and for all $g \in \chi$. Hence the scheme delivers to $K'\gamma + 1$ groups at a time, thus to

$$d_L(\gamma) = L(K'\gamma + 1) = K\gamma + L \tag{12}$$

users at a time. Then we do the same for another $\chi \in \mathcal{X}$. Along the different $\chi \in \mathcal{X}$, no subfile is repeated, and we can now conclude that the DoF is $K\gamma + L$, which as we saw (cf. (7)) is achieved here with subpacketization $S_L = \binom{K'}{K'\gamma}$.

IV. Main results

We present the main results, here for the integer case where $L|K$ and $L|K\gamma$. The interpolation to all cases K, L is easily handled using memory sharing, and it does not result in substantial performance degradation. The details for this are handled in the journal version of this work. We proceed with the main result.

Theorem 1: In the cache-aided MISO BC with L transmitting antennas and K receiving users, the delay of $T = \frac{K(1-\gamma)}{L+K\gamma}$ and the corresponding sum-DoF $d_L(\gamma) = L + K\gamma$, can be achieved with subpacketization

$$S_L = \binom{K/L}{K\gamma/L}.$$

Proof: The proof of this is direct from the description of the scheme. Specifically (7) tells us that the subpacketization is $\binom{K'}{K'\gamma}$ where $K' = K/L$, while (12) tells us that the DoF is $d_L(\gamma) = L(K'\gamma + 1) = K\gamma + L$. ■

We can now also note that the described subpacketization $S_L = \binom{K/L}{K\gamma/L}$ guarantees sum-DoF performance that is at most a factor of 2 from the theoretical optimal linear-DoF. This is direct from the bound in [14], from the performance achieved by the schemes here, and from the fact that the schemes have the 'one-shot linear' property.

A. Effective gains and multiplicative boost of effective DoF

Recall from (2) that for $L = 1$, the subpacketization takes the form $S_1 = \binom{K}{K\gamma}$, which limits the effective number of users we can encode over, from K to a smaller $\bar{K}_1 \triangleq \arg\max_{K^o \le K} \left\{\binom{K^o}{K^o\gamma} \le S_{max}\right\}$. On the other hand, in the L antenna case, the reduced subpacketization cost $S_L = \binom{\frac{K}{L}}{\frac{K\gamma}{L}}$ allows us to encode over

$$\bar{K}_L \triangleq \arg\max_{K^o \le K} \left\{\binom{\frac{K^o}{L}}{\frac{K^o\gamma}{L}} \le S_{max}\right\} = \min\{L \cdot \bar{K}_1, K\} \quad (13)$$

users. Thus going from 1 to L antennas, allows us to encode over L times as many users (up to K), which in turn offers L times more caching gain $\bar{G}_L = \min\{L \cdot \bar{G}_1, G\}$, up to the theoretical $G = K\gamma$. Specifically if $\binom{K/L}{K\gamma/L} \le S_{max}$ then $\bar{G}_L = G$ (corresponding to a multiplicative boost of $\frac{\bar{G}_L}{\bar{G}_1} = \frac{G}{\bar{G}_1}$), else the effective gain and the effective sum-DoF both experience a multiplicative increase by a factor of exactly L. For completeness this is represented in the following corollary, which follows directly from the above.

Corollary 1a: Under a maximum allowable subpacketization S_{max}, the multi-antenna effective caching gain and DoF are

$$\bar{G}_L = \min\{L \cdot \bar{G}_1, G = K\gamma\} \quad (14)$$

$$\bar{d}_L = \min\{L \cdot \bar{d}_1, d_L = L + K\gamma\} \quad (15)$$

which means that with extra antennas, the (single-antenna) effective DoF $\bar{d}_1$ is either increased by a multiplicative factor of L, or it reaches the unconstrained DoF $d_L = L + K\gamma$.

Remark 1: The L-fold multiplicative DoF boost stays into effect as long as $\binom{K/L}{K\gamma/L} \ge S_{\max}$, so in essence it stays into effect as long as subpacketization remains an issue.

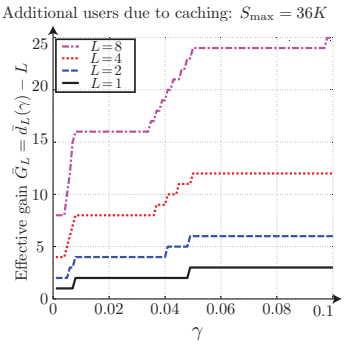

Fig. 2: Maximum achievable effective caching gain $\bar{G}_L = d_L(\gamma) - L$ (maximized over all K), of the new scheme for different L, under subpacketization constraint $S_{max} = 3.6 \cdot 10^4$.

B. Subpacketization scaling

The following corollary highlights that, in an L-antenna MISO BC system, the subpacketization cost is not determined by K or $L = \lambda K$, nor by the number of extra users G we wish to add due to caching, but rather by the ratio $x = \frac{d_L(\gamma)}{d_L(\gamma=0)}$ between the DoF and the multiplexing gain.

Corollary 1b: In our L-antenna MISO BC setting, a subpacketization of

$$S = \binom{1/\lambda}{x-1} = \binom{\frac{1}{\lambda}}{\frac{\gamma}{\lambda}}$$

can yield a DoF that is x times the multiplexing gain.

Proof: The DoF increase from $d_L(\gamma = 0) = L$ to $d_L(\gamma) = L + K\gamma = x \cdot L$, $x \in \mathbb{Z}^+$, implies that $K\gamma = L(x-1)$ and that $\gamma = \lambda(x-1)$, which means that the corresponding subpacketization $S_L = \binom{K/L}{K\gamma/L}$ now takes the form $S = \binom{\frac{1}{\lambda}}{\frac{\gamma}{\lambda}} = \binom{1/\lambda}{x-1}$. ■

Directly from the previous corollary, we also have the following.

Corollary 1c: In asymptotic terms, as long as L scales with the caching gain $K\gamma$, the entire sum-DoF $L + K\gamma$ is achievable with constant subpacketization.

Proof: As we have seen in the previous corollary, for $L = \frac{1}{q}K\gamma$ for some fixed $q \in \mathbb{Z}^+$, then the subpacketization is $S = \binom{1/\lambda}{q}$ and it is independent of K, L. ■

An additional corollary is the following.

Corollary 1d: For $L = K\gamma$, the aforementioned DoF $L + K\gamma$ can be achieved with subpacketization

$$S_L = \frac{1}{\gamma} = \frac{K}{L}.$$

The proof is direct from the above.

The following example highlights the utility of matching $K\gamma$ with L, and focuses on smaller cache sizes.

Example 1: In a BC with $\gamma = 1/100$ and $L = 1$, allowing for caching gains of $G = K\gamma = 10$ (additional users due to caching), would require $S_1 = \binom{1000}{10} > 10^{23}$ so in practice coded caching could not offer such gains. In the $L = 10$ antenna case, this caching gain comes with subpacketization of only $S_L = K/L = 100$.

V. Conclusions

In the context of coded caching with multiple transmitting antennas (or with multiple servers), we have presented a simple scheme which exploits transmitter-side dimensionality to provide very substantial reductions in the required subpacketization, without any sacrifice on the caching gain. As we have seen, this implies that, while in theory the addition of a few antennas provides an *additive* sum-DoF increase of the form

$$d_1(\gamma) = 1 + G \rightarrow d_L(\gamma) = L + G$$

(allowing to add $L-1$ extra users served at a time), in practice and in terms of subpacketization-constrained (effective) DoF, adding a few antennas implies the *multiplicative* DoF increase

$$\bar{d}_1 = 1 + \bar{G}_1 \rightarrow L + L \cdot \bar{G}_1.$$

Hence we now know that having multiple transmitting antennas, not only provides a multiplexing gain, but also a multiplicative boost of the receiver-side effective caching gain. Comparing the additive DoF increase of $L-1$ to the multiplicative DoF increase of L, suggests that the main impact of multiple transmitting antennas is not the multiplexing gain, but rather the boost on the effect of receiver-side coded caching.

A. Intuition on design

The design was based on the simple observation that multi-node (transmitter-side) precoding, reduces the need for content overlap. The subpacketization reduction from $\binom{K}{K\gamma}$ to $\binom{K/L}{K\gamma/L}$ was here related to the fact that the receivers of each group have identical caches. Subpacketization can generally increase because there needs to be a large set of pairings between the different caches. Here the number of different distinct caches is reduced, and thus the number of such pairings remains smaller.

B. Practicality and timeliness of result

The scheme consists of the basic implementable ingredients of ZF and low-dimensional coded caching, and it works for all values of K, L, γ. Its simplicity and effectiveness suggest that having extra transmitting antennas (servers) can play an important role in making coded caching even more applicable in practice, especially at a time when subpacketization complexity is the clear major bottleneck of coded caching, and also at a time when multiple antennas is a standard ingredient in wireless communications.

a) Separability between coded caching and PHY: The result also advocates that some degree of joint consideration between cache-placement and network structure (here, for receiver-side cache-placement and 'XOR' generation, we only need to know *the number* of transmitters and receivers), can yield very substantial improvements in the effective DoF. While universal coded caching schemes that work obliviously of the structure of the communication network (cf. [20]) carry an advantage when it comes to some robustness against network-structure uncertainty, the work here shows an instance where non-separated schemes have the potential to provide unboundedly better overall effective gains over universal schemes, by exploiting some of the structure of the network and by jointly considering coded caching and PHY.

b) Additional practical implications: As we argue in detail in the journal version of this work, while before we would have needed $\gamma \approx (S_{\max})^{-1/G}$ to achieve a given target gain G under a subpacketization constraint S_{max}, this is now reduced to a much smaller $\gamma \approx \left((S_{\max})^{-1/G}\right)^{L}$, further enabling smaller receiver-side caches to remain pertinent.

As we further argue in the journal version of this work, if we substitute the L-antenna array here with K_T cache-aided transmitters each with normalized cache-size γ_T, then the effective-DoF will experience a similar multiplicative boost $K_T\gamma_T$, and when this transmitter-side cache redundancy $K_T\gamma_T$ scales with $K\gamma$, then the entire sum-DoF $K_T\gamma_T + K\gamma$ is achievable with constant subpacketization.

References

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

[2] J. Zhang and P. Elia, "Wireless coded caching: A topological perspective," in *IEEE Inter. Symposium on Inform. Theory (ISIT)*. IEEE, 2017.

[3] S. S. Bidokhti, M. Wigger, and R. Timo, "Erasure broadcast networks with receiver caching," in *IEEE Inter. Symposium on Inform. Theory (ISIT)*, July 2016, pp. 1819–1823.

[4] E. Lampiris, J. Zhang, and P. Elia, "Cache-aided cooperation with no csit," in *IEEE Inter. Symposium on Inf. Theory (ISIT)*, 2017.

[5] A. Ghorbel, M. Kobayashi, and S. Yang, "Cache-enabled broadcast packet erasure channels with state feedback," in *Proc. Allerton Conf. Communication, Control and Computing*, 2015.

[6] J. Zhang and P. Elia, "Fundamental limits of cache-aided wireless BC: Interplay of coded-caching and csit feedback," *IEEE Trans. Inf. Theory*, vol. 63, no. 5, pp. 3142–3160, May 2017.

[7] J. Hachem, N. Karamchandani, and S. Diggavi, "Content caching and delivery over heterogeneous wireless networks," in *IEEE Conference on Computer Communications (INFOCOM)*, 2015.

[8] F. Engelmann and P. Elia, "A content-delivery protocol, exploiting the privacy benefits of coded caching," in *Proc. WiOpt*, May 2017, pp. 1–6.

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

[10] 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*, 2017.

[11] S.-E. Elayoubi and J. Roberts, "Performance and cost effectiveness of caching in mobile access networks," in *Proc. of the 2nd International Conference on Information-Centric Networking*, 2015.

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

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

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

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

[16] 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.

[17] J. S. P. Roig, F. Tosato, and D. Gündüz, "Interference networks with caches at both ends," *preprint arXiv:1703.04349*, 2017.

[18] S. P. Shariatpanahi, G. Caire, and B. H. Khalaj, "Multi-antenna coded caching," *preprint arXiv:1701.02979*, 2017.

[19] C. Shangguan, Y. Zhang, and G. Ge, "Centralized coded caching schemes: A hypergraph theoretical approach," *preprint arXiv:1608.03989*, 2016.

[20] N. Naderializadeh, M. A. Maddah-Ali, and A. S. Avestimehr, "On the optimality of separation between caching and delivery in general cache networks," *preprint arXiv:1701.05881*, 2017.