Low-Subpacketization Multi-Antenna Coded Caching for Dynamic Networks

MohammadJavad Salehi*, Emanuele Parrinello†, Hamidreza Bakhshzad Mahmoodi*, and Antti Tölli*

*Centre for Wireless Communications, University of Oulu, 90570 Oulu, Finland
†Communication Systems Department, Eurecom, Sophia Antipolis, 06410 Biot, France
E-mail: {firstname.lastname}@oulu.fi, {firstname.lastname}@eurecom.fr

***Abstract*—Multi-antenna coded caching combines a global caching gain, proportional to the total cache size in the network, with an additional spatial multiplexing gain that stems from multiple transmitting antennas. However, classic centralized coded caching schemes are not suitable for dynamic networks as they require prior knowledge of the number of users to indicate what data should be cached at each user during the placement phase. On the other hand, fully decentralized schemes provide comparable gains to their centralized counterparts only when the number of users is very large. In this paper, we propose a novel multi-antenna coded caching scheme for dynamic networks, where instead of defining individual cache contents, we associate users with a limited set of predefined caching profiles. Then, during the delivery phase, we aim at achieving a combined caching and spatial multiplexing gain, comparable to a large extent with the ideal case of fully centralized schemes. The resulting scheme imposes small subpacketization and beamforming overheads, is robust under dynamic network conditions, and incurs small finite-SNR performance loss compared with centralized schemes.**

***Index Terms*—coded caching, MIMO communications, low-subpacketization, dynamic networks**

I. Introduction

Wireless networks are under continuous pressure to support increasing volumes of multimedia content [1] and pave the way for the emergence of new applications such as wireless immersive viewing [2]. For the efficient delivery of such multimedia content, Maddah-Ali and Niesen proposed the idea of *coded caching* for increasing the data rates by exploiting cache content across the network [3]. In a single-stream downlink network, coded caching enables boosting the achievable rate by a multiplicative factor proportional to the cumulative cache size in the entire network. This speedup is achieved through multicasting of carefully created codewords to different groups of users, such that each user can use its cache content to remove unwanted parts from the received signal. Motivated by the growing importance of multi-antenna communications [4], cache-aided multiple-input single-output (MISO) setting was later explored in [5], [6], where it was revealed that the same coded caching gain could be achieved together with the spatial multiplexing gain. This cumulative gain was then shown in [7] to be optimal with the underlying assumptions of uncoded cache placement and one-shot linear data delivery.

Following the introduction of MISO coded caching, many later works in the literature addressed its important scaling and performance issues. Notably, in [8], the authors showed that using optimized beamformers instead of zero-forcing, one could achieve a better rate for communications at the finite-SNR regime. Similarly, the exponentially growing subpacketization issue (i.e., the number of smaller parts each file should be split into) was addressed thoroughly in the literature. Interestingly, while reducing subpacketization is challenging in single-antenna setups [9], MISO setups allow reducing the subpacketization substantially without affecting the maximum achievable degrees of freedom (DoF) [10]. Of course, this reduced subpacketization modestly decreases the achievable rate, especially at the finite-SNR regime [11]. Nevertheless, it does not harm the achievable DoF and yet enables a much simpler optimized beamformer design [12].

There exists another critical obstacle preventing the practical implementation of coded caching schemes, especially in dynamic network setups. All the schemes mentioned so far are centralized, i.e., the cache contents of the users are dictated by a central server. However, the server needs prior knowledge of the number of users to indicate what data should be cached at each user during the placement phase. This makes it impossible to implement caching techniques in dynamic networks where the users are allowed to join/leave the network at any moment. One possible solution to this issue is to use fully decentralized schemes, such as [13], where the cache contents of each user are assumed to be fully random and codewords are built to achieve the caching gain to the maximum possible extent. Unfortunately, the caching gain of such decentralized solutions is comparable with centralized schemes only asymptotically, i.e., when the number of users (or the size of each file) is very large. Hence, the problem remains largely unresolved for practical networks with a moderate number of users.

In this paper, we take a different approach to this problem by introducing a hybrid centralized/decentralized coded caching scheme. In this scheme, instead of defining cache contents for each user individually, the server assigns users with a limited set of *caching profiles* that indicate the contents that should be cached at every assigned user. Multiple users may be assigned with the same profile and cache the same contents, resulting in a shared-cache setup as studied in [14]. However, due to

This work is supported by the Academy of Finland under grant no. 318927 (6Genesis Flagship), and by Vaikuttavuussäätiö under the project Directional Data Delivery for Wireless Immersive Digital Environments (3D-WIDE).

the dynamic nature of the network, the *length* of each profile, defined as the number of users associated with that profile, can vary during the time and be independent from the length of other profiles. Then, we introduce a new delivery algorithm, inspired by the low-subpacketization scheme in [12], for enabling a combined caching and spatial multiplexing gain, comparable to a large extent with the ideal case of fully centralized schemes. Of course, this paper is not the first one introducing such a hybrid scheme; a similar approach is introduced in [15] for single-antenna setups. However, here, we extended the results to multi-antenna setups and do so with minor subpacketization and beamforming overheads.

We emphasize that, this paper does *not* aim at proposing an information-theoretic DoF-optimal scheme for dynamic networks. Instead, the goal is to provide a practical scheme with an appropriate performance at the finite-SNR regime. In this regard, we leave most of the theoretical analyses of the proposed scheme to the extended version of this paper, and use numerical simulations here to investigate how our new scheme performs in comparison with centralized schemes applied to static network setups. Nevertheless, we provide helpful insights about the underlying reasons for any observation of an improved/deteriorated performance.

In this paper, we use boldface lower- and upper-case letters to denote vectors and matrices, respectively. Sets are shown with calligraphic letters. Similar to vectors, it is assumed that the order of members in a set is important and preserved. $\mathbf{A}[i,j]$ is the element at row i and column j of matrix $\mathbf{A}$, and $\mathbf{a}[i]$ and $\mathcal{A}[i]$ are the i-th elements of vector $\mathbf{a}$ and set $\mathcal{A}$, respectively. $[K]$ represents the set of integers $\{1,2,...,K\}$, and $\mathcal{A}\backslash\mathcal{B}$ denotes the elements of set $\mathcal{A}$ which are not in set $\mathcal{B}$. Other notations are defined as they are used in the text.

II. System Model

We consider a MISO dynamic setup where a group of cache-enabled single-antenna users request data from a multi-antenna server with the spatial multiplexing gain of α.[1] The users request data from a library $\mathcal{F}$ of size N files, where the size of each file is f bits. Every user has a cache memory large enough to store a portion $0 < \gamma < 1$ of the entire library. We define the parameter P to be the smallest integer such that $P\gamma$ is also an integer. For example, if $\gamma = \frac{1}{4}$, then, $P = 4$.

The users are allowed to join and leave the network at any time. Upon joining the network, every user k is assigned (e.g., randomly) with a profile index $\mathsf{p}(k) \in [P]$. The profile index indicates what contents should be cached at the cache memory of a user (more detailed explanation is provided shortly after). After being assigned with the profile index $\mathsf{p}(k)$, user k starts filling up its cache memory by downloading data from the server. We assume this cache filling process does not affect simultaneously ongoing data delivery processes.[2] After all the required data is downloaded in the cache memory of user k, this user is identified by the server to be eligible to participate in the upcoming coded caching (CC) data delivery phase.

Data requests are revealed to the server at specific time intervals. At the beginning of each time interval, a subset of users reveal their requested data from the library $\mathcal{F}$.[3] We denote the file requested by user k as $W(k)$. The server then carries out data delivery in two subsequent phases:

1) **CC data delivery**, where the server builds a set of transmission vectors using multi-antenna coded caching techniques and transmits them to eligible users (that have filled their cache memories with the intended data);
2) **Unicast data delivery**, where the server transmits all the remaining requested data, not delivered in the CC phase.

The first phase is designed to enable a combined global caching and spatial multiplexing gain, comparable with the ideal case of fully centralized schemes. However, in the unicast phase, only the local caching gain is achieved together with the spatial DoF. In this paper, without loss of generality, we consider only a specific request interval and drop the interval index. We assume the whole process is repeated at every time interval after a new subset of users reveal their requests.

The CC delivery phase is based on the RED scheme in [12]. This scheme allows achieving the largest possible combined DoF of caching and spatial multiplexing with a very small subpacketization requirement. It also enables implementing high-performance optimized beamformers with very low complexity (using uplink-downlink duality). The same technique is used in this paper to design optimized beamformers for both delivery phases. As discussed earlier, we explore only the finite-SNR performance here, and leave theoretical analysis for the extended version of this paper. In this regards, we use the total required delivery time (over both phases) as our metric. Symmetric rate is then defined as the inverse of the total delivery time.

Cache Placement. The content cached at each user k is defined by its assigned profile $\mathsf{p}(k)$. Multiple users may be assigned with the same profile, and hence, cache the same content. In order to indicate which data elements should be stored for each profile, we use a $P \times P$ binary placement matrix $\mathbf{V}$. The first row of $\mathbf{V}$ has $\bar{t} = P\gamma$ consecutive ones (other elements are zero), and the row $1 < p \leq P$ is a circular shift of the row $p-1$ to the right by one unit. Given $\mathbf{V}$, we split each file $W \in \mathcal{F}$ into P equal-sized *packets* W_p. Then, if $\mathbf{V}[p, \mathsf{p}(k)] = 1$ for some user k and profile index p, user k is instructed to cache packets W_p of every file $W \in \mathcal{F}$.

[1] The real number of antennas may be larger than the spatial DoF. However, we assume the spatial DoF at the users and the server is *set* to one and α, respectively. In [8], it is thoroughly studied how spatial DoF value affects the performance of coded caching schemes at finite- and high-SNR regimes.

[2] This is the case, for example, if the cache is filled when excess transmission capacity becomes available at the server (e.g., when the network traffic load is low), or if the users join the network from specific locations with large transmission capacities (e.g., when transmit antennas are located near the entrance door of the application area in an extended reality use case).

[3] The reason for assuming that a *subset* of users request data is that in some applications, the requests are made only when the user conditions are changed. For example, consider an extended reality application where the requests are made only when the users move to a different location.

Example 1. *Consider a MISO setup where $\gamma = \frac{1}{4}$, and hence, $P = 4$ and $\bar{t} = 1$. Then, the placement matrix is*

$$\mathbf{V} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} .$$

So, each file is split into $P = 4$ packets, and, for example, if $\mathsf{p}(k) = 1$ for some user k, then it has to cache packet W_1 from every file $W \in \mathcal{F}$.

III. CC Delivery Phase

After the users reveal their requests, the server first performs a CC delivery phase where it uses coded caching techniques to build a set of codewords and transmit them to eligible users. Assume the number of users associated with profile p and making requests at the considered time instant (i.e., the length of profile p) is η_p. The server then chooses a common parameter $\hat{\eta}$ to indicate the number of users participating in the CC delivery phase. The system level impact of choosing $\hat{\eta}$ is clarified later in this section. Then, for every profile p:

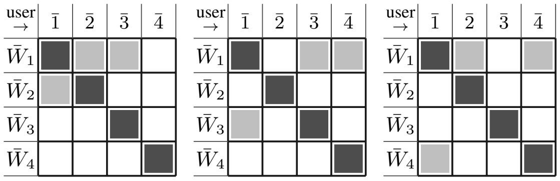
1) if $\eta_p > \hat{\eta}$, then $\hat{\eta} - \eta_p$ users associated with profile p are selected randomly and excluded from the CC phase;
2) if $\eta_p \leq \hat{\eta}$, then $\hat{\eta} - \eta_p$ *phantom* users are assigned with the profile p.

Phantom users are non-existing users that appear when codewords are built and are excluded from the actual delivery [12]. Although phantom users incur some DoF loss, they enable a larger beamforming gain that can actually improve the achievable rate, compared with schemes with a larger DoF, at the finite-SNR regime (cf. [8], [12]).

After the above-mentioned process, all profiles will have the same length of $\hat{\eta}$. Then, we consider a *virtual* network with P users, coded caching gain $\bar{t} = P\gamma$, and spatial DoF of $\bar{\alpha} = \lceil \frac{\alpha}{\hat{\eta}} \rceil$, where the cache contents of the virtual user $\bar{k} \in [P]$ is the same as a real user k with $\mathsf{p}(k) = \bar{k}$. For this virtual network, we build the transmission vectors using the RED scheme in [12].[4] This results in P transmission rounds where at each round $P - \bar{t}$ transmissions are done. Denoting the j-th transmission vector at round r with $\bar{\mathbf{x}}_j^r$, we have

$$\bar{\mathbf{x}}_j^r = \sum_{\bar{n} \in [\bar{t}+\bar{\alpha}]} \bar{W}_{\bar{\mathbf{p}}_j^r[\bar{n}]}^{\bar{q}}(\bar{\mathbf{k}}_j^r[\bar{n}])\bar{\mathbf{w}}_{\bar{\mathcal{R}}_j^r(\bar{n})} , \tag{1}$$

where $\bar{W}(\bar{k})$ denotes the file requested by the virtual user $\bar{k}$, $\bar{q}$ is the subpacket index initialized to one and increased every time a packet appears in a transmission vector, and $\bar{\mathbf{w}}_{\bar{\mathcal{R}}_j^r(\bar{n})}$ is the optimized beamforming vector suppressing data at every virtual user in the *interference indicator* set $\bar{\mathcal{R}}_j^r(\bar{n})$. Moreover, $\bar{\mathbf{p}}_j^r$ and $\bar{\mathbf{k}}_j^r$ are the packet and user index vectors, respectively, built such that the graphical representation of the transmission vectors follows two perpendicular circular shift operations over a grid (cf. [12]). The following example clarifies how the transmission vectors for the virtual network are built.



(a) $\bar{\mathbf{x}}_1^1$ (b) $\bar{\mathbf{x}}_2^1$ (c) $\bar{\mathbf{x}}_3^1$

Fig. 1: Graphical representation of transmission vectors for the virtual network

Example 2. *Consider the network in Example 1, and assume the spatial DoF is $\alpha = 4$. Also, assume $\eta_1 = \eta_2 = 2$ and $\eta_3 = \eta_4 = 3$. If we choose either $\hat{\eta} = 2$ or $\hat{\eta} = 3$, the virtual network will have $P = 4$ users, coded caching gain $\bar{t} = 1$ and spatial DoF $\bar{\alpha} = 2$. Then, data delivery for the virtual network is done in four rounds, with three transmissions at each round. According to [12], user and packet index vectors for the first round are*

$$\bar{\mathbf{k}}_1^1 = [\bar{1},\bar{2},\bar{3}], \quad \bar{\mathbf{k}}_2^1 = [\bar{1},\bar{3},\bar{4}], \quad \bar{\mathbf{k}}_3^1 = [\bar{1},\bar{4},\bar{2}] ,$$
$$\bar{\mathbf{p}}_1^1 = [\bar{2},\bar{1},\bar{1}], \quad \bar{\mathbf{p}}_2^1 = [\bar{3},\bar{1},\bar{1}], \quad \bar{\mathbf{p}}_3^1 = [\bar{4},\bar{1},\bar{1}] ,$$

resulting in transmission vectors

$$\begin{aligned} \bar{\mathbf{x}}_1^1 &= \bar{W}_{\bar{2}}^{\bar{1}}(\bar{1})\bar{\mathbf{w}}_{\bar{3}} + \bar{W}_{\bar{1}}^{\bar{1}}(\bar{2})\bar{\mathbf{w}}_{\bar{3}} + \bar{W}_{\bar{1}}^{\bar{1}}(\bar{3})\bar{\mathbf{w}}_{\bar{2}} , \\ \bar{\mathbf{x}}_2^1 &= \bar{W}_{\bar{3}}^{\bar{1}}(\bar{1})\bar{\mathbf{w}}_{\bar{4}} + \bar{W}_{\bar{1}}^{\bar{2}}(\bar{3})\bar{\mathbf{w}}_{\bar{4}} + \bar{W}_{\bar{1}}^{\bar{1}}(\bar{4})\bar{\mathbf{w}}_{\bar{3}} , \\ \bar{\mathbf{x}}_3^1 &= \bar{W}_{\bar{4}}^{\bar{1}}(\bar{1})\bar{\mathbf{w}}_{\bar{2}} + \bar{W}_{\bar{1}}^{\bar{2}}(\bar{4})\bar{\mathbf{w}}_{\bar{2}} + \bar{W}_{\bar{1}}^{\bar{2}}(\bar{2})\bar{\mathbf{w}}_{\bar{4}} , \end{aligned} \tag{2}$$

where brackets for interference indicator sets are dropped for notational simplicity. For clarification, we use the tabular view in Figure 1, borrowed from [12], to graphically represent the transmission vectors. In this representation, columns and rows denote user and packet indices, respectively. A darkly shaded cell means the packet index is cached at the user, while a lightly shaded cell indicates that (part of) the packet index is transmitted to the user. As can be seen in Figure 1, the transmission vectors in a single round are built using circular shift operations of the lightly shaded cells over non-colored cells of the table, in two perpendicular directions (in column one from up to down and in row one from left to right).

After the transmission vectors are built for the virtual network, we need to *elevate* them to be applicable to the original network. We first need to define a few new parameters; b is the remainder of the division of α by $\hat{\eta}$, i.e., $b = \alpha - \hat{\eta}\lfloor \frac{\alpha}{\hat{\eta}} \rfloor$, and the set $\mathcal{D}_{\bar{k}}$ consists of the set of user indices k in the real network for which $\mathsf{p}(k) = \bar{k}$. Then, we have two elevation mechanisms, depending on the value of b.

1) If $b = 0$, i.e., if $\frac{\alpha}{\hat{\eta}}$ is an integer, every vector $\bar{\mathbf{x}}_j^r$ is elevated into one transmission vector $\mathbf{x}_j^r$ for the original network. To do so, each term inside the summation in (1) is replaced by

$$\bar{W}_{\bar{\mathbf{p}}_j^r[\bar{n}]}^{\bar{q}}(\bar{\mathbf{k}}_j^r[\bar{n}])\bar{\mathbf{w}}_{\bar{\mathcal{R}}_j^r(\bar{n})} \rightarrow \sum_{m \in [\hat{\eta}]} W_{\bar{\mathbf{p}}_j^r[\bar{n}]}^{q}(\mathcal{D}_{\bar{\mathbf{k}}_j^r[\bar{n}]}[m])\mathbf{w}_{\mathcal{R}_j^r(\bar{n},m)} ,$$

where

$$\mathcal{R}_j^r(\bar{n},m) = \bigcup_{\bar{k} \in \bar{\mathcal{R}}_j^r(\bar{n})} \mathcal{D}_{\bar{k}} \ \cup \ \mathcal{D}_{\bar{\mathbf{k}}_j^r[\bar{n}]} \ \setminus \ \{\mathcal{D}_{\bar{\mathbf{k}}_j^r[\bar{n}]}[m]\} . \tag{3}$$

In other words, every data term intended for some user $\bar{k}$ in the virtual network is replaced by $\hat{\eta}$ data terms intended for all real users $k \in \mathcal{D}_{\bar{k}}$, and the inter-stream interference between these

[4] The RED scheme requires the spatial DoF to be larger than or equal to the caching gain. So, here we assume $\bar{\alpha} > \bar{t}$. Relaxing this condition is left for the extended version of the paper.

terms is suppressed by beamforming vectors. Note that each interference indicator set $\bar{\mathcal{R}}_j^r(\bar{n})$ in the virtual network consists of $\bar{\alpha}-1$ virtual users, and hence, using (3), every interference indicator set in the real network will have $(\bar{\alpha}-1)\hat{\eta}+(\hat{\eta}-1)=\bar{\alpha}\hat{\eta}-1=\alpha-1$ users. So, suppressing the interference with the beamformers is possible as the spatial DoF for the original network is α. Summarizing the discussions, if $b=0$, the transmission vectors for the original network are built as

$$\mathbf{x}_j^r = \sum_{\bar{n}\in[\bar{t}+\bar{\alpha}]} \sum_{m\in[\hat{\eta}]} W^q_{\bar{\mathbf{p}}_j^r[\bar{n}]}(\mathcal{D}_{\bar{\mathbf{k}}_j^r[\bar{n}]}[m])\mathbf{w}_{\mathcal{R}_j^r(\bar{n},m)} \ . \tag{4}$$

2) If $b \neq 0$, i.e., when α is *not* divisible by $\hat{\eta}$, every transmission vector $\bar{\mathbf{x}}_j^r$ is elevated into $\hat{\eta}$ transmission vectors $\mathbf{x}_{j,s}^r$, $s\in[\hat{\eta}]$, for the original network. For elevation, we first define the *base* and *extended* interference indicator sets as

$$\mathcal{B}_{j,s}^{r,b}(\bar{n},m) = \bigcup_{\bar{k}\in\bar{\mathcal{R}}_j^r[\bar{n}]} \mathcal{D}_{\bar{k}} \bigcup_{i\in\mathcal{E}_s^b(\hat{\eta})} \{\mathcal{D}_{\bar{\mathbf{k}}_j^r[\bar{t}+\bar{\alpha}]}[i]\}$$

and

$$\mathcal{R}_{j,s}^{r,b}(\bar{n},m)=\begin{cases}\mathcal{B}_{j,s}^{r,b}(\bar{n},m)\cup\mathcal{D}_{\bar{\mathbf{k}}_j^r[\bar{n}]}\backslash\{\mathcal{D}_{\bar{\mathbf{k}}_j^r[\bar{n}]}[m]\} & \bar{n}<\bar{t}+\bar{\alpha}\\ \mathcal{B}_{j,s}^{r,b}(\bar{n},m)\backslash\{\mathcal{D}_{\bar{\mathbf{k}}_j^r[\bar{n}]}[m]\} & \bar{n}=\bar{t}+\bar{\alpha}\end{cases}$$

respectively, where $\mathcal{E}_s^b(\hat{\eta})$ denotes the first b elements of $[\hat{\eta}]$, after they are circularly shifted to the left by s units (recall that the order of members in a set is preserved). In other words,

$$\mathcal{E}_s^b(\hat{\eta}) = \big(([b]+s-2) \mod \hat{\eta}\big)+1 \ . \tag{5}$$

Then, the elevated transmission vectors are built as

$$\mathbf{x}_{j,s}^r = \sum_{\bar{n}\in[\bar{t}+\bar{\alpha}]} \sum_{m\in[\hat{\eta}]} W^q_{\bar{\mathbf{p}}_j^r[\bar{n}]}(\mathcal{D}_{\bar{\mathbf{k}}_j^r[\bar{n}]}[m])\mathbf{w}_{\mathcal{R}_{j,s}^{r,b}(\bar{n},m)} \ . \tag{6}$$

As an explanation, from the $\bar{t}+\bar{\alpha}$ total data terms within the transmission vector $\bar{\mathbf{x}}_j^r$, we substitute each of the first $\bar{t}+\bar{\alpha}-1$ terms with $\hat{\eta}$ terms for the real network (as we did for the case $b=0$). However, the last data term is only replaced by b terms, out of the $\hat{\eta}$ options, chosen through a circular shift operation (implemented using $\mathcal{E}_s^b(\hat{\eta})$ in the equations). The following example clarifies the elevation process for both cases of $b=0$ and $b\neq 0$, and enlightens the effect of phantom users.

Example 3. *Consider the same network in Example 1, with the profile lengths mentioned in Example 2. Let us assume* $\mathcal{D}_{\bar{1}}=[1,2]$, $\mathcal{D}_{\bar{2}}=[3,4]$, $\mathcal{D}_{\bar{3}}=[5,6,7]$, *and* $\mathcal{D}_{\bar{4}}=[8,9,10]$. *For this network, we find transmission vectors resulting from the elevation of the virtual transmission vectors* $\bar{\mathbf{x}}_1^1$ *and* $\bar{\mathbf{x}}_1^2$ *in* (2), *for both cases of* $\hat{\eta}=2,3$.

➢ $\hat{\eta}=2$*: In this case, we need no phantom users but should exclude two users from* $\mathcal{D}_{\bar{3}}$ *and* $\mathcal{D}_{\bar{4}}$ *from the CC delivery phase. Assume users 7 and 10 are excluded. As* $b=0$, *we can use* (4) *to elevate* $\bar{\mathbf{x}}_1^1$ *and* $\bar{\mathbf{x}}_1^2$ *as*

$$\begin{aligned}\mathbf{x}_1^1 =& W_2^1(1)\mathbf{w}_{5,6,2}+W_2^1(2)\mathbf{w}_{5,6,1}+W_1^1(3)\mathbf{w}_{5,6,4}\\ &+W_1^1(4)\mathbf{w}_{5,6,3}+W_1^1(5)\mathbf{w}_{3,4,6}+W_1^1(6)\mathbf{w}_{3,4,5} \ ,\\ \mathbf{x}_2^1 =& W_3^1(1)\mathbf{w}_{8,9,2}+W_3^1(2)\mathbf{w}_{8,9,1}+W_1^2(5)\mathbf{w}_{8,9,6}\\ &+W_1^2(6)\mathbf{w}_{8,9,5}+W_1^1(8)\mathbf{w}_{5,6,9}+W_1^1(9)\mathbf{w}_{5,6,8} \ ,\end{aligned}$$

and their graphical representation will be as shown in Figure 2. Comparing Figures 1 and 2, we can see that the elevation mechanism stretches the transmission vectors horizontally.

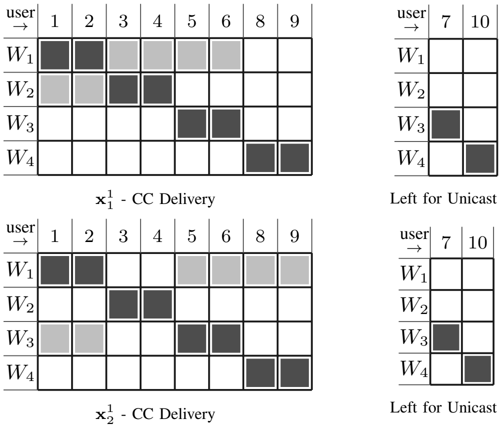


Fig. 2: Elevated versions of $\bar{\mathbf{x}}_1^1$ (top) and $\bar{\mathbf{x}}_2^1$ (bottom) - The case $b=0$

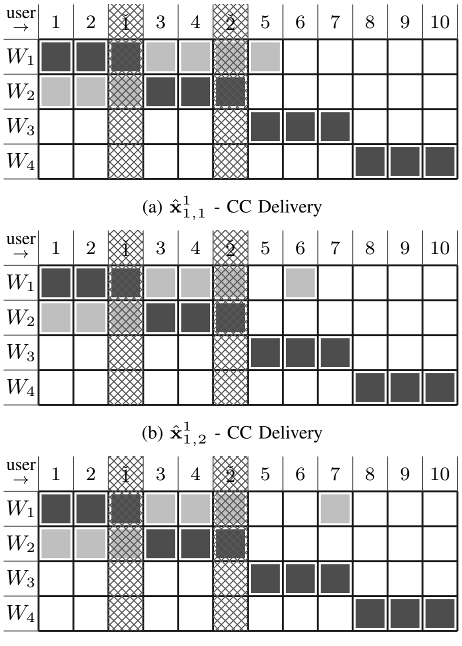
➢ $\hat{\eta}=3$*: In this case, no user is excluded from the CC delivery phase but two phantom users* $\check{1}$ *and* $\check{2}$ *should be added to* $\mathcal{D}_{\bar{1}}$ *and* $\mathcal{D}_{\bar{2}}$*, respectively. As* $b=1$, *each transmission vector for the virtual network is elevated into* $\hat{\eta}=3$ *vectors for the real network. Here we only mention the vectors resulting from* $\bar{\mathbf{x}}_1^1$*, which are built as*

$$\begin{aligned}\hat{\mathbf{x}}_{1,1}^1 =& W_2^1(1)\mathbf{w}_{2,\check{1},5}+W_2^1(2)\mathbf{w}_{1,\check{1},5}+W_2^1(\check{1})\mathbf{w}_{1,2,5}+\\ &W_1^1(3)\mathbf{w}_{4,\check{2},5}+W_1^1(4)\mathbf{w}_{3,\check{2},5}+W_1^1(\check{2})\mathbf{w}_{3,4,5}+W_1^1(5)\mathbf{w}_{3,4,\check{2}},\\ \hat{\mathbf{x}}_{1,2}^1 =& W_2^2(1)\mathbf{w}_{2,\check{1},6}+W_2^2(2)\mathbf{w}_{1,\check{1},6}+W_2^2(\check{1})\mathbf{w}_{1,2,6}+\\ &W_1^2(3)\mathbf{w}_{4,\check{2},6}+W_1^2(4)\mathbf{w}_{3,\check{2},6}+W_1^2(\check{2})\mathbf{w}_{3,4,6}+W_1^1(6)\mathbf{w}_{3,4,\check{2}},\\ \hat{\mathbf{x}}_{1,3}^1 =& W_2^3(1)\mathbf{w}_{2,\check{1},7}+W_2^3(2)\mathbf{w}_{1,\check{1},7}+W_2^3(\check{1})\mathbf{w}_{1,2,7}+\\ &W_1^3(3)\mathbf{w}_{4,\check{2},7}+W_1^3(4)\mathbf{w}_{3,\check{2},7}+W_1^3(\check{2})\mathbf{w}_{3,4,7}+W_1^1(7)\mathbf{w}_{3,4,\check{2}}.\end{aligned} \tag{7}$$

The graphical representations of the elevated vectors are shown in Figure 3, where the columns representing phantom users are hatched for better clarification. Comparing with the base transmission vectors for the virtual network in Figure 1, the elevation mechanism works through horizontal stretching followed by an extra circular shift operation over a block of user indices (users 5,6,7 in this example). Finally, we should note that the effect of phantom users should be removed before the real transmission. To do so, we simply remove phantom indices from the transmission vectors. So, for example, instead of $\hat{\mathbf{x}}_{1,1}^1$ *in* (7), *we transmit*

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

As can be seen in Example 3, the choice of $\hat{\eta}$ can heavily affect both performance and complexity metrics such as the required subpacketization and the number of transmissions. A smaller $\hat{\eta}$ reduces the chance of requiring phantom users, but more real users are excluded from the CC delivery phase, and hence, the achievable coded caching gain is reduced. On the other hand, while with a larger $\hat{\eta}$ we can include more users in the CC delivery phase, in specific transmissions, the achievable DoF can become too small after the effect of phantom users is removed (in Example 3, $\mathbf{x}_1^1$ has a DoF of six, but for $\hat{\mathbf{x}}_{1,s}^1$, $s\in[3]$ the actual DoF is reduced to five). In fact, if the distribution of profile lengths η_p and the choice of $\hat{\eta}$ are such



(a) $\hat{\mathbf{x}}_{1,1}^1$ - CC Delivery

(b) $\hat{\mathbf{x}}_{1,2}^1$ - CC Delivery

(c) $\hat{\mathbf{x}}_{1,3}^1$ - CC Delivery

Fig. 3: Elevated versions of $\bar{\mathbf{x}}_1^1$ - The case $b \neq 0$

that we need many phantom users, the achievable DoF in some transmissions may fall below the spatial DoF α (i.e., we lose DoF by using coded caching techniques). One solution to this issue is to check the achievable DoF at every transmission vector after removing the effect of phantom users, and avoid the transmission if the DoF falls below α. The data terms intended to be sent with such transmission vectors are then transmitted during the subsequent unicast delivery phase.

IV. Unicast Delivery Phase

During the unicast delivery phase, we fulfill the request of users excluded from the CC delivery phase. Meantime, we also transmit any other data term not sent during the CC delivery phase. This includes, for example, data terms in transmission vectors with a DoF value smaller than α (due to the presence of phantom users), or data terms re-transmitted because of channel errors. We use a greedy mechanism (which is not necessarily optimal) to achieve an appropriate spatial DoF for every transmission vector in this phase. To do so, we first split the files requested by users excluded from the CC delivery phase into the same number of subpackets we needed in the CC delivery phase (this subpacketization value is discussed shortly after). Then, for every user k, we define $u(k)$ to be the number of subpackets that should be delivered to user k, and sort users by their $u(k)$ values in the descending order. Now, we select $\min\{\alpha, U\}$ first users, where U is the total number of users for which $u(k) > 0$, and deliver one subpacket to each selected user, with a single transmission vector. The inter-stream interference in the transmission vector is suppressed by beamforming vectors (which is possible as the spatial DoF is α). Finally, we update $u(k)$ values and repeat the same procedure, until all the remaining data terms are transmitted.

Example 4. *Consider the network in Example 3, and assume $\hat{\eta} = 2$. In this case, we don't need any phantom users, but the users 7 and 10 are excluded from the CC delivery phase (as also shown in Figure 2). As will be shown later, the total subpacketization in this case is $P(\bar{t}+\bar{\alpha}) = 12$. However, in the placement phase, each file is already split into $P = 4$ packets, and one packet is cached at each user. So, during the delivery, we need to further split each packet into $\frac{12}{4} = 3$ subpackets and deliver nine subpackets to each of these users. So, $u(k) = 9$ for $k = 7, 10$ and zero otherwise, and $U = 2$. Then, we need nine transmission vectors, where each vector delivers one subpacket to each of the users 7 and 10. For example, the first and second vectors are built as $W_1^1(7)\mathbf{w}_{10} + W_1^1(10)\mathbf{w}_7$ and $W_1^2(7)\mathbf{w}_{10} + W_1^2(10)\mathbf{w}_7$, respectively.*

V. Performance Analysis

As stated earlier, the goal of this paper is *not* to provide a theoretically optimal scheme. Instead, we aim at proposing a practical coded caching solution for dynamic networks where the users are allowed to join and leave the network. In this regard, our scheme has a small subpacketization requirement and enables a very simple optimized beamformer design using uplink-downlink duality [12], and hence, can be applied to networks with a large ($O(10^2)$) number of users. Also, as we will verify through simulations, it is able to provide a comparable performance with the non-dynamic scenario.

The subpacketization requirement of our scheme depends on the value of b. If $b = 0$ (i.e., if α is divisible by $\hat{\eta}$), the total subpacketization is $P(\bar{t}+\bar{\alpha})$. This simply follows the fact that in this case, every transmission vector for the virtual network is elevated into exactly one vector for the original network, and no extra splitting is required. However, if $b \neq 0$, the scheme needs a larger subpacketization of $P(\hat{\eta}\bar{t} + \alpha)$. Analyzing the required subpacketization in this case is lengthy due to the extra circular shift operation in the scheme and removed here due to the lack of space. It can be seen that in both cases, the required subpacketization is pretty small and grows linearly with the total number of users in the CC delivery phase.

Simulation results for the proposed scheme are provided in Figures 4 and 5. Simulations are performed for a MISO network with 50 single-antenna users, where the cache ratio at each user is $\gamma = 0.1$ (i.e., $P = 10$), and the spatial DoF is set to $\alpha = 10$. The number of transmitting antennas at the server can be any number larger than α; it is assumed to be 12 here. For the distribution of η_p, we consider uniform distribution (i.e., $\eta_p = 5$ for $p \in [10]$) as well as three other scenarios, where the non-uniformity of η_p values increase from scenario 1 to 3. The exact values of η_p for these scenarios are shown in Table I. In all simulations, optimized beamformers are used and both CC and unicast delivery phases are implemented. For a better comparison, in the figures we have also included the rate curve for the case all the required data is delivered through the unicast delivery phase (i.e., there is no coded caching gain).

In Figure 4, the effect of the $\hat{\eta}$ parameter on the performance in scenarios 1, 3 is shown. In general, choosing a larger $\hat{\eta}$ value (up to $\max \eta_p$) improves the performance. This is because

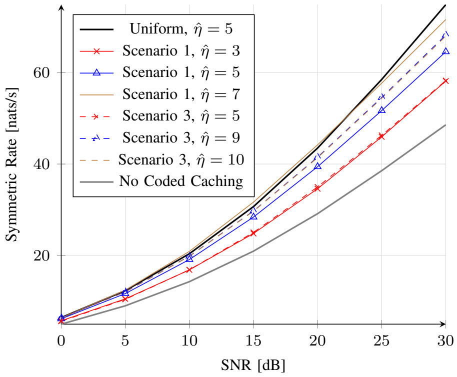


Fig. 4: Performance comparison for various $\hat{\eta}$ values, scenarios 1 and 3

Scenario	η_p values	Standard deviation
1	5,4,5,5,4,3,6,6,5,7	1.15
2	9,3,1,4,5,7,2,6,5,8	2.58
3	8,3,8,0,4,10,7,4,0,6	3.40

TABLE I: Simulation scenarios with non-uniform profile lengths

with increased $\hat{\eta}$, a better coded caching gain, strong enough to cover for the DoF loss of phantom users, is achieved. Of course, it should be noted that in scenario 3, the case $\hat{\eta} = 9$ provides slightly (about one percent) better rate compared with $\hat{\eta} = 10$. This is because as η_p distribution becomes more non-uniform, by choosing $\hat{\eta} = \max \eta_p$ we need to add too many phantom users, increasing the chance of DoF loss. Nevertheless, even for the very non-uniform case of scenario 3, this performance loss is negligible, and one can safely consider $\hat{\eta} = \max \eta_p$ in all cases. Of course, a more theoretical analysis, due for the extended version of this paper, would better clarify the probable performance loss of selecting $\hat{\eta} = \max \eta_p$.

Finally, in Figure 5, we compare the performance of the proposed scheme with the case of uniform η_p distribution (i.e., the maximum possible coded caching gain), for scenarios 1, 2, and 3, when $\hat{\eta} = \max \eta_p$. As can be seen, the proposed scheme performs very well, with loss of less than ten percent for the considered network setup, compared with the uniform case. This is because with $\hat{\eta} = \max \eta_p$, we try to maximize the achievable global caching gain. Also, when DoF falls due to phantom users, optimized beamformers enable a larger beamforming gain (as the size of the interference indicator set becomes smaller than $\alpha - 1$), thus compensating for the minor performance loss. Interestingly, in the low-SNR regime and for scenario 1, the proposed scheme even outperforms the uniform case by a small margin. This is because the beamforming gain is more effective in this regime, as discussed in [8].

VI. Conclusion and Future Work

We introduced a new multi-antenna coded caching scheme for dynamic networks where the users are allowed to join and leave the network freely. Instead of dictating individual caching profiles for users, we associated them with a set of predefined caching profiles. Data delivery is then performed in two consecutive phases, with the goal of maximizing the achievable performance. The proposed scheme requires a small subpacketization, enables easy implementation of optimized beamformers, and its performance and robustness under dynamic conditions is shown through simulations. The current paper provides a proof-of-concept, with many theoretical analyses and optimizations due for future work.

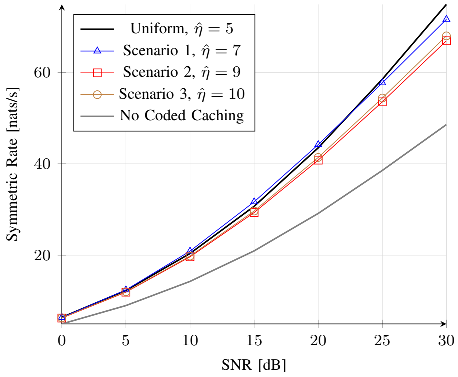


Fig. 5: Performance comparison for $\hat{\eta} = \max \eta_p$, scenarios 1, 2, and 3

References

[1] E. Summary, “Cisco Visual Networking Index – Forecast and,” *Europe*, vol. 1, pp. 2007–2012, 2012.

[2] H. B. Mahmoodi, M. J. Salehi, and A. Tolli, “Non-Symmetric Coded Caching for Location-Dependent Content Delivery,” *IEEE International Symposium on Information Theory - Proceedings*, vol. 2021-July, pp. 712–717, 2021.

[3] M. A. Maddah-Ali and U. Niesen, “Fundamental limits of caching,” *IEEE Transactions on Information Theory*, vol. 60, no. 5, pp. 2856–2867, 2014.

[4] N. Rajatheva *et al.*, “White paper on broadband connectivity in 6g,” *arXiv preprint arXiv:2004.14247*, 2020.

[5] S. P. Shariatpanahi, S. A. Motahari, and B. H. Khalaj, “Multi-server coded caching,” *IEEE Transactions on Information Theory*, vol. 62, no. 12, pp. 7253–7271, 2016.

[6] S. P. Shariatpanahi, G. Caire, and B. Hossein Khalaj, “Physical-Layer Schemes for Wireless Coded Caching,” *IEEE Transactions on Information Theory*, vol. 65, no. 5, pp. 2792–2807, 2019.

[7] E. Lampiris and P. Elia, “Resolving a Feedback Bottleneck of Multi-Antenna Coded Caching,” *arXiv*, 2018. [Online]. Available: http://arxiv.org/abs/1811.03935

[8] A. Tölli, S. P. Shariatpanahi, J. Kaleva, and B. H. Khalaj, “Multi-antenna interference management for coded caching,” *IEEE Transactions on Wireless Communications*, vol. 19, no. 3, pp. 2091–2106, 2020.

[9] Q. Yan, X. Tang, Q. Chen, and M. Cheng, “Placement Delivery Array Design Through Strong Edge Coloring of Bipartite Graphs,” *IEEE Communications Letters*, vol. 22, no. 2, pp. 236–239, 2018.

[10] E. Lampiris and P. Elia, “Adding transmitters dramatically boosts coded-caching gains for finite file sizes,” *IEEE Journal on Selected Areas in Communications*, vol. 36, no. 6, pp. 1176–1188, 2018.

[11] M. Salehi, A. Tolli, S. P. Shariatpanahi, and J. Kaleva, “Subpacketization-rate trade-off in multi-antenna coded caching,” in *2019 IEEE Global Communications Conference, GLOBECOM 2019 - Proceedings*. IEEE, 2019, pp. 1–6.

[12] M. J. Salehi, E. Parrinello, S. P. Shariatpanahi, P. Elia, and A. Tolli, “Low-Complexity High-Performance Cyclic Caching for Large MISO Systems,” *IEEE Transactions on Wireless Communications*, vol. 21, no. 5, pp. 3263–3278, 2022.

[13] M. A. Maddah-Ali and U. Niesen, “Decentralized Coded Caching Attains Order-Optimal Memory-Rate Tradeoff,” *IEEE/ACM Transactions on Networking*, vol. 23, no. 4, pp. 1029–1040, 2015.

[14] E. Parrinello, A. Unsal, and P. Elia, “Fundamental Limits of Coded Caching with Multiple Antennas, Shared Caches and Uncoded Prefetching,” *IEEE Transactions on Information Theory*, vol. 66, no. 4, pp. 2252–2268, 2020.

[15] S. Jin, Y. Cui, H. Liu, and G. Caire, “A new order-optimal decentralized coded caching scheme with good performance in the finite file size regime,” *IEEE Transactions on Communications*, vol. 67, no. 8, pp. 5297–5310, 2019.