Asymmetric Multi-Antenna Coded Caching for Location-Dependent Content Delivery

Hamidreza Bakhshzad Mahmoodi, MohammadJavad Salehi, and Antti Tölli

Centre for Wireless Communications, University of Oulu, 90570 Oulu, Finland
E-mail: {firstname.lastname}@oulu.fi

***Abstract*—Efficient usage of in-device storage and computation capabilities are key solutions to support data-intensive applications such as immersive digital experiences. This paper proposes a location-dependent multi-antenna coded caching - based content delivery scheme tailored specifically for wireless immersive viewing applications. First, a novel memory allocation process incentivizes the content relevant to the identified wireless bottleneck areas. This enables a trade-off between local and global caching gains and results in unequal fractions of location-dependent multimedia content cached by each user. Then, a novel packet generation process is carried out during the subsequent delivery phase, given the asymmetric cache placement. During this phase, the number of packets transmitted to each user is the same, while the sizes of the packets are proportional to the corresponding location-dependent cache ratios. In this regard, each user is served with location-specific content using joint multicast beamforming and a multi-rate modulation scheme that simultaneously benefits from global caching and spatial multiplexing gains. Numerical experiments and mathematical analysis demonstrate significant performance gains compared to the state-of-the-art.**

***Index Terms*—Multi-antenna communications, coded caching, location-dependent content delivery, immersive viewing.**

I. Introduction

It is expected that 5G penetration will pass the ten percent mark by 2023. The average per-user throughput will go beyond 570 Mbps [1], more than a ten-fold increase compared with what was achievable five years earlier with 4G-LTE. This is primarily due to new data-intensive services such as wireless immersive viewing offered by 5G and beyond [2]–[5]. Indeed, supporting the high-data-rate wireless connectivity required by such data-intensive applications necessitates more advanced solutions than merely increasing the available bandwidth [3]. Meanwhile, improving caching and computing capabilities at end-users has been deemed highly effective to increase the transmission efficiency [5], [6]. As such, upcoming mobile broadband applications rely heavily on asynchronous content reuse [7], and hence, proactive caching of popular content at the end-users could relieve network congestion and bandwidth consumption during peak traffic demand times [8].

The coded caching (CC) technique, initially proposed by Maddah-Ali and Niesen in [9], has gained attention during the last years due to an additional *global caching gain* compared to traditional (local) caching schemes. Remarkably, the global caching gain scales linearly with the total number of users in the network, making it appealing for multi-user collaborative use cases such as immersive viewing application [10]. Furthermore, an exciting property of CC schemes is their capability of combining global caching and spatial multiplexing gains resulting from multi-antenna transmissions [11].

However, since multi-antenna CC schemes are generally based on multicasting [11], the achievable rate in any multicast message is limited to the rate of the user with the worst channel condition, which is known as *near-far* issue. In fact, authors in [12] show that the effective gain of the CC scheme could entirely vanish at low-SNR for the single-input-single-output (SISO) setting. To address this issue, several approaches exist (e.g., [13]–[15]), among which the nested code modulation - based (NCM) scheme, proposed in [15], is more appealing as it creates multicast messages that serve every user in a multicasting group with a different rate. The multi-rate property in [15] is achieved by altering the modulation constellation using side information available at each user (c.f. [16]–[18]). However, despite this great effort by the research community, the near-far issue still needs to be addressed for dynamic real-time applications where users frequently move within the network and their achievable rate changes accordingly.

In this paper, a new multi-antenna CC scheme is introduced for efficient content delivery in wireless access networks with location-dependent content requests. This work is a major extension of our earlier paper [4] now applied to multi-antenna setups. We consider a wireless connectivity scenario where the users are free to move and their requested contents depend on their instantaneous locations. A multi-user immersive viewing environment is considered as a specific use-case, where several users are submerged into a network-based immersive application that runs on high-end eyewears. Such a use case entails a substantial volume of multimedia traffic with guaranteed quality of experience (QoE) throughout the operating environment. To this end, similar to [4], a location-dependent uneven memory allocation process is first carried out. However, unlike [4] which only considers the local caching gain to minimize the estimated delivery time, we jointly consider global caching and multiplexing gains on top of local caching gain to reflect the achievable degrees of freedom (DoF) by the

This work is supported by the Academy of Finland under grants no. 318927 (6G Flagship), 319059 and 343586, and by the Finnish Research Impact Foundation (Vaikuttavuussäätiö) under the project Directional Data Delivery for Wireless Immersive Digital Environments (3D-WIDE).

proposed delivery scheme. This process results in an uneven cache placement, causing the users to have distinct cache ratios and conventional CC delivery schemes to *not* be applicable anymore. Therefore, a novel packet generation scheme is devised to handle the irregularity by creating packets with sizes proportional to the corresponding uneven cache ratios. Finally, a multicast beamforming scheme with an underlying multi-rate modulation is proposed to leverage aggregate global caching and multiplexing gains simultaneously, improving the QoE compared to the state-of-the-art.

Notations: Boldface lower-case letters denote vectors and calligraphic letters represent sets. $\mathcal{A}\backslash\mathcal{B}$ is the set of elements in $\mathcal{A}$ which are not in $\mathcal{B}$. Also, $|\mathcal{A}|$ and $\|\mathbf{v}\|$ denote the cardinality of set $\mathcal{A}$ and the norm of vector $\mathbf{v}$, respectively.

II. System Model

We envision a bounded environment (game hall, operating theatre, etc.) where a server with L transmit antennas serves K single-antenna users[1] through wireless communication links. The set of users is denoted by $\mathcal{K} = \{1, \ldots, K\}$. The users are equipped with finite-size cache memories and are free to move throughout the environment. Every user requests data from the server at each time slot based on the application's needs and its location. The requested data content can be divided into static and dynamic parts, where the former can be proactively stored in the user cache memories. This paper focuses on the wireless delivery of this static location-dependent content part, partially aided by in-device cache memories.[2] A real-world application of such a setup is a wireless immersive digital experience environment, where the requested data is needed to reconstruct the location-dependent 3D field-of-view (FoV) at each user. Naturally, users in different locations experience different channel conditions due to varying wireless connectivity. Thus, the goal is to design a cache-aided communication scheme that minimizes the maximum delivery time and provides as uniform QoE as possible, irrespective of the users' locations.

Intuitively, a larger share of the total cache memory should be reserved for storing data needed in locations with poor communication quality. In this regard, the environment is split into S regions such that all points in a given region have almost the same level/quality of wireless connectivity. In other words, regions are small enough such that channel variation among different points in a given region is negligible. In the following, we refer to these regions as *states* and denote the set of states as $\mathcal{S}$. A graphical example of an application environment with its states is provided in figure 1. The file required for reconstructing the FoV of state $s \in \mathcal{S}$ is denoted by $W(s)$, and without loss of generality, we assume that the size of $W(s)$ is F bits for all states. Moreover, every user has a cache memory of size MF bits. If not stated otherwise, we consider a normalized data unit in the following and drop F in subsequent notations.

[1]The system model can be easily extended to multi-antenna receivers following a similar approach as proposed in [19].

[2]We assume that a portion of the achievable data rate available at each user is dedicated to deliver the dynamic content without cache assistance.

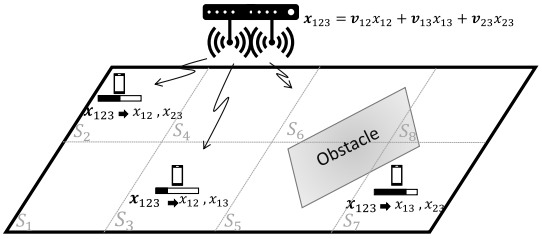

Fig. 1: An application environment with $K = 3$ users is split into $S = 8$ states, where users $k = \{1, 2, 3\}$ are located in states $s_k = \{3, 2, 7\}$, respectively. State-specific expected rate used for cache placement is represented by $r(s)$, where $r(3) > r(2) > r(7)$. The transmitted message $\mathbf{x}_{123}$ consist of $x_{\mathcal{U}}$ and $\{\mathbf{v}_{\mathcal{U}}, \mathcal{U} \ni k\}$ representing the data part intended for users $k \in \mathcal{U}$ and the corresponding multicast precoder, respectively. The black bar below each user indicates how much of the requested data is cached.

We assume instantaneous channel knowledge is available at the transmitter and used for beamformer design and rate allocation during the delivery phase. However, for the location-dependent cache placement phase, we need a prior estimate of the achievable rate at different states, for which we use a hypothetical single-user scenario for convenience (note that actual user-specific delivery rates depend on the number of users scheduled in parallel, precoding algorithms used, etc.). As a result, the expected interference-free spectral efficiency attained in state $s \in \mathcal{S}$ can be simply approximated as

$$\bar{r}(s) = \mathbb{E}\big[\log(1 + \frac{P_T \ \|\mathbf{h}_s\|^2}{N_0})\big] \quad \text{[bits/s/Hz]}, \tag{1}$$

where P_T is the transmission power, N_0 is the additive white Gaussian noise power, and $\mathbf{h}_s \in \mathbb{C}^L$ is the channel vector between the server and a user located in state s.[3] Note that the expectation is taken over all the realizations in time and space/locations in state s. We consider a wideband communication scheme, where the total bandwidth is divided into several small frequency bins. Thus, the expected data rate over all the frequency bins is approximated as $\hat{r}(s) \sim B\bar{r}(s)$, where B is the communication bandwidth. For ease of exposition, we consider normalised data rate $r(s) = \frac{\hat{r}(s)}{F}$ and assume $F = B$, throughout this paper. Moreover, we present the delivery procedure for a specific time slot and ignore the time index (the same procedure is repeated at each slot).

III. Cache Placement

The placement phase is executed, for example, before the users enter the application environment or when they pass through specific high-data-rate locations (data shower). During this phase, users' cache memories are filled up with valuable data aimed to minimize the duration of the upcoming content delivery phase. Here, we use a similar procedure as in [4] for the location-dependent cache placement phase. Thus, the placement phase comprises two consecutive processes, *memory allocation* and *cache arrangement*. While the cache arrangement process is the same as [4], the memory allocation process is tailored to reflect the multi-antenna support.

Memory Allocation: Due to the considered real-time application, it is crucial to guarantee data delivery within a limited

[3]While we use (1) for convenience, the expected location-specific data rates can be attained through various means, e.g., via collecting statistics from past active users.

time. Intuitively, this requires reserving a larger share of the total cache memory for storing data needed in locations with poor communication quality. In this regard, the amount of cache memory dedicated for storing (parts of) every state-specific content file $W(s)$ at each user is determined during this process. In this paper, we assume that there is no a priori knowledge about the users' spatial locations during the delivery phase, and hence, use uniform access probability for all the states (using priory knowledge about the states popularity, the performance can be improved). Let us use $m(s)$ to denote the normalized cache size at each user allocated to store (parts of) $W(s)$. Since the size of $W(s)$ is normalized to one, a user in state s needs to receive $1-m(s)$ data units over the wireless link to reconstruct the FoV of state s. Now, to capture the effect of our multi-antenna multicasting delivery scheme presented in section IV on the memory allocation process, we leverage the following lemma.

Lemma 1. *If $m(s)$ values are known, the total delivery time of the proposed scheme, denoted by T_T, is approximated as*

$$\hat{T}_T = \frac{K}{\bar{t}+L} \max_{s\in\mathcal{S}} \frac{1-m(s)}{r(s)}\ , \tag{2}$$

where $\bar{t} = K\bar{m}$ is the least common cache ratio of all users, $r(s)$ is the approximated rate at state s and $\bar{m} = \min\limits_{k\in[K]} m(s_k)$.

Proof. Due to lack of space, the proof is left for the extended version of the paper (c.f. [20]). □

Now, we first rewrite (2) as $\hat{T}_T = \frac{1}{\bar{m}+\frac{L}{K}} \max_{s\in\mathcal{S}} \frac{1-m(s)}{r(s)}$, and then, to minimize the delivery time for any possible realizations of user locations, we formulate the memory allocation process as the following linear fractional programming (LFP):

$$\begin{aligned} &\min_{m(s),\ \gamma\geq 0,\ m\geq 0} \quad \frac{\gamma}{m+\frac{L}{K}} \\ &\text{s.t.} \quad \frac{1-m(s)}{r(s)} \leq \gamma,\ \forall s\in\mathcal{S}, \\ &m \leq m(s),\ \forall s\in\mathcal{S}, \quad \sum_{s\in\mathcal{S}} m(s) \leq M. \end{aligned} \tag{3}$$

Note that at the optimal point, $m = \bar{m} = \min_{s\in[S]} m(s)$. Using Charnes-Cooper transformation [21], this problem can be reformulated as an equivalent linear programming (LP) as:

$$\begin{aligned} &\min_{m^{'}(s),\ \gamma^{'}\geq 0,\ m^{'}\geq 0\ \xi\geq 0} \quad \gamma^{'}, \\ &\text{s.t.} \quad \frac{\xi - m^{'}(s)}{r(s)} \leq \gamma^{'},\ \forall s\in\mathcal{S},\ m^{'} + \frac{L}{K}\xi = 1, \\ &m^{'} \leq m^{'}(s),\ \forall s\in\mathcal{S}, \quad \sum_{s\in\mathcal{S}} m^{'}(s) \leq M\xi. \end{aligned} \tag{4}$$

Problem (4) is convex and can be solved efficiently. After solving this problem, the actual allocated memory would be $m(s) = m^{'}(s)/\xi,\ \forall s$. Compared to the memory allocation process proposed in [4], here, the term $\bar{m}+\frac{L}{K}$ is considered in (3) to reflect the multi-antenna multi-user support, resulting in improved overall performance as shown in Section V. This improvement is due to considering the multiplicative effect of the common cache ratio $\bar{t}$ in (4), where we avoid drastically decreasing minimum allocated memory while increasing local caches at the bottleneck areas, resulting in higher achievable DoF compared to [4].

Algorithm 1 Location-based cache placement

```
procedure CACHE_PLACEMENT
    {m(s)} = The result of the LFP problem in (3)
    for all s ∈ S do
        t(s) = K × m(s)
        W(s) → {W_{V(s)}(s) | V(s) ⊆ K, |V(s)| = t(s)}
        for all V(s) do
            for all k ∈ K do
                if k ∈ V(s) then
                    Put W_{V(s)}(s) in the cache of user k
```

	$s=1$	$s=2$	$s=3$	$s=4$	$s=5$
Expected rate $r(s)$	3	2	1	2	3
Allocated memory $m(s)$	0.25	0.5	0.75	0.5	0.25

TABLE I: Location-specific rate and memory allocation for Example 1.

Remark 1. Substituting the term $\frac{L}{K}$ in (3) with a general constant term ϕ enables a trade-off between the local and global caching gains. Selecting a large $\phi \gg \frac{L}{K}$ prioritizes the local caching gain $m(s)$ at the expense of the minimum global caching gain $\bar{t} = K\bar{m}$ (as the denominator in the objective function becomes almost constant). On the other hand, if $\phi \ll \frac{L}{K}$, the minimum allocated memory $\bar{m}$ converges to $\frac{M}{S}$ and the minimum global caching gain is maximized at the cost of lower local caching gain for the states with poor expected connectivity $r(s)$, resulting in higher QoE fluctuations.

Cache Arrangement. After the memory allocation process, we store data in the cache memories of the users following a similar method as proposed in [9]. In this regard, for every state $s\in\mathcal{S}$, we first split $W(s)$ into $\binom{K}{t(s)}$ sub-files denoted by $W_{\mathcal{V}(s)}(s)$, where $t(s) = Km(s)$ and $\mathcal{V}(s)$ can be any subset of the user-set $\mathcal{K}$ with $|\mathcal{V}(s)| = t(s)$. Then, at the cache memory of user $k\in\mathcal{K}$, we store $W_{\mathcal{V}(s)}(s)$ for every state $s\in\mathcal{S}$ and set $\mathcal{V}(s) \ni k$. The cache arrangement process is outlined in Algorithm 1. We assume for every $s\in\mathcal{S}$, $m(s) > 0$ and $t(s)$ is an integer. Analyzing the case these constraints are not met is left for the extended version of this paper (c.f. [20]). Finally, for notational simplicity, we ignore the brackets and separators while explicitly writing $\mathcal{V}(s)$, e.g., $W_{ij}(s) \equiv W_{\{i,j\}}(s)$.

Example 1. *Consider an immersive viewing application with $K = 4$ users, where the environment is split into $S = 5$ states and for each state, the required data size is $F = 400$ [MB].Each user has a cache size of 900MB, and hence, the normalized cache size is $M = 2.25$ data units. The spatial distribution of the approximated normalized rate and the corresponding memory allocation for each state s after solving* (3) *are as shown in Table I. It can be easily verified that $t(1) = t(5) = 1$, $t(2) = t(4) = 2$, and $t(3) = 3$. As a result, $W(1)$, $W(3)$ and $W(5)$ should be split into 4 sub-files, while $W(2)$ and $W(4)$ are split into $\binom{4}{2} = 6$ sub-files. The resulting cache placement is visualized in Figure 2.*

IV. Content Delivery

At the beginning of the delivery phase, every user $k\in\mathcal{K}$ reveals its requested file $W_k \equiv W(s_k)$, i.e., W_k depends on

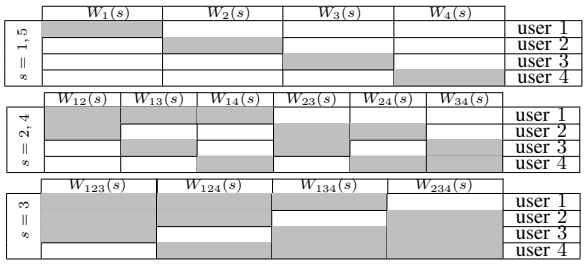

Fig. 2: Cache placement visualization for Example 1.

the state s_k where user k is located. The server then builds and transmits several *nested* codewords[4], such that after receiving the codewords, all the targeted users can reconstruct their requested files. User k requires a total amount of one normalized data unit to reconstruct W_k, as detailed in Section II. However, only a subset of this data, with size $m_k \equiv m(s_k)$, is available in its cache. Therefore, the remaining part should be delivered by the server. Note that the conventional multi-server CC-based delivery scheme, as used in [11], assumes all users have the same *cache ratio*. Hence, it does not apply to our considered scenario where each user has cached a different ratio of its requested file. Thus, a new delivery mechanism is required to achieve a proper multicasting gain.

The new delivery algorithm is outlined in Algorithm 2. First, the server builds and transmits multiple transmission vectors $\mathbf{x}_{\bar{\mathcal{K}}}$ in a time division multiple access (TDMA) manner for every subset of users $\bar{\mathcal{K}} \subseteq \mathcal{K} : |\bar{\mathcal{K}}| = \hat{t} + L$, where $\hat{t}$ is the *common cache ratio* defined as $\hat{t} = \min_{k\in\mathcal{K}} t_k$, and $t_k \equiv t(s_k)$. The transmitted signal vector

$$\mathbf{x}_{\bar{\mathcal{K}}} = \sum_{\mathcal{U}\subseteq\bar{\mathcal{K}},|\mathcal{U}|=\hat{t}+1} \mathbf{v}_{\mathcal{U}} x_{\mathcal{U}} \ , \tag{5}$$

is comprised of multiple unit-power nested codewords $x_{\mathcal{U}}$, where $\mathcal{U}$ can be any subset of $\bar{\mathcal{K}}$ with $|\mathcal{U}| = \hat{t} + 1$. Also, every nested codeword $x_{\mathcal{U}}$ is precoded with a tailored beamformer vector $\mathbf{v}_{\mathcal{U}} \in \mathbb{C}^L$, designed to suppress (or null-out) the interference caused by $x_{\mathcal{U}}$ on every user $\bar{k} \in \bar{\mathcal{K}} \setminus \mathcal{U}$. Here we use optimized beamformers as they provide a better performance especially at the finite-SNR regime [11]. The optimized multicast beamformers can be obtained by solving the following weighted max-min optimization

$$\begin{aligned} \underset{\{\mathbf{v}_{\mathcal{U}}\}_{k\in\hat{\mathcal{K}}}}{\text{maxmin}} \ \underset{\substack{\mathcal{Q}\subseteq\mathcal{D}_k,\\ |\mathcal{Q}|\neq 0}}{\min} \ & \frac{1}{|Y_{\mathcal{U},k}||\mathcal{Q}|} \log\left(1+\frac{\sum_{\mathcal{U}\in\mathcal{Q}}|\mathbf{h}_k^H\mathbf{v}_{\mathcal{U}}|^2}{\sum_{\mathcal{V}\in\mathcal{I}_k}|\mathbf{h}_k^H\mathbf{v}_{\mathcal{V}}|^2+N_0}\right) \\ \text{s.t.} \ & \sum_{\mathcal{U}\subseteq\hat{\mathcal{K}},|\mathcal{U}|=\hat{t}+1} \|\mathbf{v}_{\mathcal{U}}\|^2 \le P_T, \end{aligned} \tag{6}$$

where, for user k, $\mathcal{D}_k := \{\mathcal{U} \mid \mathcal{U} \subseteq \hat{\mathcal{K}}, |\mathcal{U}| = \hat{t}+1, \mathcal{U} \ni k\}$ and $\mathcal{I}_k := \{\mathcal{V} \mid \mathcal{V} \subseteq \hat{\mathcal{K}} \setminus k, |\mathcal{V}| = \hat{t}+1\}$ are the set of all desired and interfering message indices in $\mathbf{x}_{\hat{\mathcal{K}}}$, respectively. Also, $Y_{\mathcal{U},k}$ represents the data term/packet for user k in the nested codeword $x_{\mathcal{U}}$ (explained in more details shortly after), and $|Y_{\mathcal{U},k}|$ is the length of $Y_{\mathcal{U},k}$ in bits used for relative weighting in (6). Note that (6) is a modified version of max-min problem in [11] (equations (37)-(41)) with additional weights to reflect the multi-rate modulation transmission. Hence, problem (6) is non-convex and NP-hard, which can be sub-optimally solved using the successive convex approximation method [11]. However, a detailed discussion is left for the extended version of this paper due to lack of space (c.f. [20]).

[4]Here we consider the NCM scheme [15] to support multi-rate transmission. However, the scheme is oblivious to the modulation procedure and any other multi-rate modulation scheme may be used (e.g., [16]–[18]).

Algorithm 2 NCM-based Content Delivery

1: **procedure** DELIVERY
2: $\hat{t} = \min_{k\in\mathcal{K}} t_k$
3: **for all** $\bar{\mathcal{K}} \subseteq \mathcal{K} : |\bar{\mathcal{K}}| = \hat{t} + L$ **do**
4: $\mathbf{x}_{\bar{\mathcal{K}}} \leftarrow 0$
5: **for all** $\mathcal{U} \subseteq \bar{\mathcal{K}} : |\mathcal{U}| = \hat{t} + 1$ **do**
6: $x_{\mathcal{U}} \leftarrow 0$
7: **for all** $k \in \mathcal{U}$ **do**
8: $\alpha_k \leftarrow \binom{t_k}{\hat{t}}\binom{K-\hat{t}-1}{L-1}$, $Y_{\mathcal{U},k} \leftarrow 0$, $\mathcal{U}_{-k} \leftarrow \mathcal{U}\backslash\{k\}$
9: **for all** $\mathcal{V}_k \subseteq \bar{\mathcal{K}} : |\mathcal{V}_k| = t_k + 1$ **do**
10: **if** $\mathcal{U}_{-k} \subseteq \mathcal{V}_k$, $k \notin \mathcal{V}_k$ **then**
11: $W^q_{\mathcal{V}_k,k} \leftarrow$ CHUNK$(W_{\mathcal{V}_k,k}, \alpha_k)$
12: $Y_{\mathcal{U},k} \leftarrow$ CONCAT $(Y_{\mathcal{U},k}, W^q_{\mathcal{V}_k,k})$
13: $x_{\mathcal{U}} \leftarrow$ NEST $(x_{\mathcal{U}}, Y_{\mathcal{U},k}, R_k)$
14: $\mathbf{x}_{\bar{\mathcal{K}}} \leftarrow \mathbf{x}_{\bar{\mathcal{K}}} + \mathbf{v}_{\mathcal{U}} x_{\mathcal{U}}$
15: Transmit $\mathbf{x}_{\bar{\mathcal{K}}}$

The nested codeword $x_{\mathcal{U}}$ is built to include a useful data term/packet $Y_{\mathcal{U},k}$ for every user $k \in \mathcal{U}$. The data term $Y_{\mathcal{U},k}$ is chosen to be available in the cache memory of every other user $\bar{k} \in \mathcal{U} \setminus \{k\}$, so that these users can remove its interference using their cache contents. To satisfy this condition, denoting $\mathcal{U}_{-k} \equiv \mathcal{U} \setminus \{k\}$, we build $Y_{\mathcal{U},k}$ to include (parts of) every *suitable* sub-file $W_{\mathcal{V}(s_k),k}$ for which $\mathcal{U}_{-k} \subseteq \mathcal{V}(s_k)$ and $k \notin \mathcal{V}(s_k)$. However, since $W_{\mathcal{V}(s_k),k}$ is cached in the cache memory of every user $\bar{k} \in \mathcal{V}(s_k)$ and $|\mathcal{U}_{-k}| = \hat{t} \le t_k = |\mathcal{V}(s_k)|$, we may find more than one suitable sub-file $W_{\mathcal{V}(s_k),k}$ to be included in $Y_{\mathcal{U},k}$. In fact, there exist exactly

$$\beta_k = \binom{K-\hat{t}-1}{t_k-\hat{t}}$$

suitable sub-files for inclusion in $Y_{\mathcal{U},k}$, which should be split into smaller parts and *concatenated* while building $x_{\mathcal{U}}$. Note that every sub-file $W_{\mathcal{V}(s_k),k}$ appears in $\binom{t_k}{\hat{t}}$ different $\mathcal{U}_{-k}$ sets (c.f. step 10 in Algorithm 2), and each user set $\mathcal{U}$ is targeted $\binom{K-\hat{t}-1}{L-1}$ times during the delivery phase (c.f. steps 3 and 5). Hence, to send fresh content in each transmission, we need to divide every sub-file $W_{\mathcal{V}(s_k),k}$ suitable for user k into exactly

$$\alpha_k = \binom{t_k}{\hat{t}}\binom{K-\hat{t}-1}{L-1}$$

equal-sized segments (denoted by $W^q_{\mathcal{V}(s_k),k}$ in Algorithm 2) before the concatenation. In other word, we split every suitable sub-file into α_k segments, and then concatenate β_k number of these segments to build $Y_{\mathcal{U},k}$.

The function CHUNK in Algorithm 2 ensures none of the segments of a sub-file is sent twice, and the function CONCAT creates a bit-wise concatenation of the given segments. The final codeword $x_{\mathcal{U}}$ is then created by nesting $Y_{\mathcal{U},k}$ for every user $k \in \mathcal{U}$, shown by the auxiliary function NEST. Using the nesting operation (c.f. [18]) to create codeword $x_{\mathcal{U}}$, and beamformers $\mathbf{v}_{\mathcal{U}}$ in (6), we are able to simultaneously transmit every $Y_{\mathcal{U},k}$ with rate R_k, such that $\frac{|Y_{\mathcal{U},k}|}{R_k} = \frac{|Y_{\mathcal{U},i}|}{R_i}, \forall (k,i) \in \mathcal{U}$.

Example 2. *Consider the network in Example 1, for which the cache placement is visualized in Figure 2. Assume there exist two antennas at the transmitter (i.e., $L = 2$). Let us consider a specific time slot, in which $s_1 = 1$, $s_2 = 2$, $s_3 = 4$, $s_4 = 5$. Denoting the set of requested sub-files for user k with $\mathcal{T}_k$ and assuming $A \equiv W(1)$, $B \equiv W(2)$, $C \equiv W(4)$, and $D \equiv W(5)$, we have*

$$\begin{aligned} &\mathcal{T}_1 = \{A_2, A_3, A_4\}, \qquad \mathcal{T}_2 = \{B_{13}, B_{14}, B_{34}\}, \\ &\mathcal{T}_3 = \{C_{12}, C_{14}, C_{24}\}, \quad \mathcal{T}_4 = \{D_1, D_2, D_3\}. \end{aligned} \tag{7}$$

Note that, the size of the sub-files of A, B, C, D are $\frac{1}{4}, \frac{1}{6}, \frac{1}{6}, \frac{1}{4}$ data units, respectively. As $L = 2$ and the common cache ratio is $\hat{t} = 1$, our proposed algorithm can deliver data to $\hat{t}+L = 3$ users during each transmission. Let us consider the transmission vector $\mathbf{x}_{123}$ for users $\bar{\mathcal{K}} = \{1, 2, 3\}$. Following equation (5)*, we have $\mathbf{x}_{123} = \mathbf{v}_{12}x_{12}+\mathbf{v}_{13}x_{13}+\mathbf{v}_{23}x_{23}$, where the nested codewords x_{12}, x_{13}, and x_{23} deliver a portion of the requested data to user sets $\{1, 2\}$, $\{1, 3\}$ and $\{2, 3\}$, respectively. Based on the users' request sets $\mathcal{T}_k$, there exist only one suitable sub-file for user* 1*, whereas, for user* 2 *and* 3*, there exist two (i.e., $\beta_1 = 1$ and $\beta_2 = \beta_3 = 2$). We also need to split sub-files into smaller segments using user-specific factors $\alpha_1 = 2$ and $\alpha_2 = \alpha_3 = 4$.*

Hence, x_{12} is built as $x_{12} = A_2^1 * \prod(B_{13}^1, B_{14}^1)$, where the operator $(*)$ denotes the nesting operation and $\prod(A, B)$ represents the bit-wise concatenation of data segments A and B (superscripts are used to differentiate various segments of a sub-file). The nesting operation in x_{12} is performed such that A_2^1 and $\prod(B_{13}^1, B_{14}^1)$ are delivered with proportional rates $R_1 = \frac{3}{2} * R_2$. Following the same procedure to build x_{13} and x_{23}, the transmission vector $\mathbf{x}_{123}$ is formed as

$$\begin{aligned} \mathbf{x}_{123} =& \mathbf{v}_{12}\Big(A_2^1 * \prod(B_{13}^1, B_{14}^1)\Big) + \mathbf{v}_{13}\Big(A_3^1 * \prod(C_{12}^1\ C_{14}^1)\Big) \\ &+ \mathbf{v}_{23}\Big(\prod(B_{13}^2, B_{34}^1) * \prod(C_{12}^2, C_{24}^1)\Big). \end{aligned}$$

Now, let us consider the decoding process for $\mathbf{x}_{123}$ at user 1*. For notational simplicity, let us ignore the interference terms suppressed by beamforming vectors. Then, user* 1 *receives*

$$y_1 = \underline{A_2^1 * \prod(B_{13}^1, B_{14}^1)}\mathbf{h}_1^H\mathbf{v}_{12} + \underline{A_3^1 * \prod(C_{12}^1, C_{14}^1)}\mathbf{h}_1^H\mathbf{v}_{13} + w_1,$$

where w_1 is the white additive noise at user one. To decode its requested data terms A_2^1 and A_3^1, user 1 *has to jointly decode the two underlined messages (using successive interference cancellation [11]), benefiting from its cache contents (i.e., $\prod(B_{13}^1, B_{14}^1)$ and $\prod(C_{12}^1, C_{14}^1)$) as receiver side* a priori *knowledge for demodulation. Similarly, users two and three can also decode their requested data terms interference-free.*

Lemma 2. *Using the proposed cache placement and content delivery algorithms, every user receives its requested data.*

Proof. The user k in state s_k needs to receive $1 - m_k$ data units during the delivery phase. This data is delivered by $\binom{K-1}{\hat{t}+L-1}$ transmission vectors $\mathbf{x}_{\hat{\mathcal{K}}}$ for which $\hat{\mathcal{K}} \ni k$. However, the number of nested codewords $x_{\mathcal{U}}$ for which $\mathcal{U} \ni k$ in every such vector $\mathbf{x}_{\hat{\mathcal{K}}}$ is $\binom{\hat{t}+L-1}{\hat{t}}$, and each $x_{\mathcal{U}}$ delivers a data term $Y_{\mathcal{U},k}$ to user k that is comprised of β_k segments each with size $1/\alpha_k\binom{K}{t_k}$ data units. Hence, the total data size delivered to user k is $\frac{\binom{K-1}{\hat{t}+L-1}\binom{\hat{t}+L-1}{\hat{t}}\binom{K-\hat{t}-1}{t_k-\hat{t}}}{\binom{K}{t_k}\binom{t_k}{\hat{t}}\binom{K-\hat{t}-1}{L-1}} = \frac{K-t_k}{K} = 1 - m_k$. □

Intuitively, for every transmission vector $\mathbf{x}_{\hat{\mathcal{K}}}$ we should have $\frac{|Y_{\mathcal{U},k}|}{R_k} = \frac{|Y_{\mathcal{U},i}|}{R_i}$, $\forall k, i \in \hat{\mathcal{K}}$, where R_k is the dedicated transmission rate to user k (c.f. [20] for details). This simply results in

$$\frac{R_k}{1 - m_k} = \frac{R_i}{1 - m_i}, \qquad \forall k, i \in \bar{\mathcal{K}}. \tag{8}$$

Note that R_k is equal to $\frac{1}{\binom{\hat{t}+L-1}{\hat{t}}}\log\left(1 + \sum_{\mathcal{U}\in\mathcal{D}_k}\gamma_{\mathcal{U}}^k\right)$ at the optimal point [20], where $\gamma_{\mathcal{U}}^k = \frac{|\mathbf{h}_k^H\mathbf{v}_{\mathcal{U}}|^2}{\sum_{\mathcal{V}\in\mathcal{I}_k}|\mathbf{h}_k^H\mathbf{v}_{\mathcal{V}}|^2+N_0}$. As a result, (8) can be written as $\frac{r_k}{1-m_k} = \frac{r_i}{1-m_i}$, $\forall k, i \in \bar{\mathcal{K}}$, where $r_k = \log\left(1 + \sum_{\mathcal{U}\in\mathcal{D}_k}\gamma_{\mathcal{U}}^k\right)$ is the sum-rate. Now, let us define $R_{\mathrm{w}} = \frac{r_k}{1-m_k}$ as the common weighted rate. Then, the transmission time for vector $\mathbf{x}_{\bar{\mathcal{K}}}$ would be $T_{\bar{\mathcal{K}}} = 1/\binom{K-1}{\hat{t}+L-1}R_{\mathrm{w}}$, which is independent of users' states. Assuming R_{w} is almost the same for any subset of users $\bar{\mathcal{K}}$, the total delivery time can be approximated as $T_T \approx \binom{K}{\hat{t}+L}/\binom{K-1}{\hat{t}+L-1}R_{\mathrm{w}} = K/(\hat{t}+L)R_{\mathrm{w}}$, and the symmetric rate will be $R_{\mathrm{w}}^s = \frac{K}{T_T} = (\hat{t}+L)R_{\mathrm{w}}$. Following a similar argument for the multiantenna CC scheme in [11], the symmetric rate would be $R_u^s = \frac{K}{T_T} = (t+L)R_u$, where $R_u = \frac{\bar{r}}{1-M/S}$, $\bar{r}$ is the common max-min sum-rate, and $t = KM/S$ is the common caching gain (c.f. [11] Section IV).

V. Numerical results

We use numerical simulations to evaluate the performance of the proposed location-dependent scheme. We consider a bounded 5×5 [m^2] room divided into $S = 121$ states, where a transmitter with $L = 2$ antennas is located in the middle of the room on the ceiling at 3 [m] height. The channel at state $s \in \mathcal{S}$ is modelled as $\mathbf{h}_s \sim \mathbb{CN}(\mathbf{0}, \sigma_s d_s^{-\eta}\mathbf{I})$, where d_s is the distance between the center of the state s and the transmitter, $\eta = 3$ is the pass-loss exponent and σ_s is the shadowing effect. To reflect the shadowing effect, we assume Q states (out of S states) are highly attenuated (e.g., located behind obstacles like walls, etc. see Fig. 1). We assume a simple binary attenuation model for σ_s, where $\sigma_s = 1$ for non-attenuated states and $\sigma_s = \xi$ for attenuated ones. We assume the transmit power is such that the received signal-to-noise ratio (SNR) at the non-attenuated states at the room borders is equal to P [dB]. For a user k located in state s during the delivery phase, we assume $\mathbf{h}_k = \mathbf{h}(s)$. At each time-slot, users $k = 1, \ldots, K$ are placed in any state $s = 1, \ldots, S$ with uniform probability. For comparison, we use both the total transmission time to serve all the users T_T and symmetric rate, defined as $\frac{K}{T_T}$, averaged over all channel realizations. In all the simulations, we use optimized beamformers obtained by solving (6).

The proposed scheme is compared with three benchmarks. **Unicast** refers to uniform cache allocation followed by unicast data delivery (i.e., serving L users during each transmission), enabling spatial multiplexing and local caching gains only. **Multiserver** refers to uniform cache allocation and conventional CC delivery scheme of [11]. Finally, **LD-multicast** is

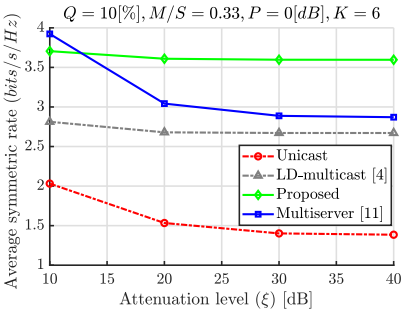

Fig. 3: Rate versus attenuation intensity (ξ).

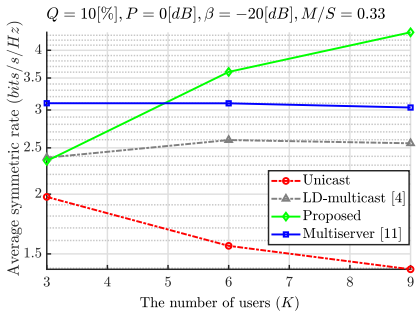

Fig. 4: Rate versus different number of users (K).

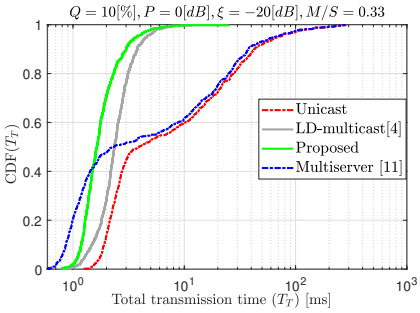

Fig. 5: CDF of the total transmission time (T_T).

a direct extension of [4], where the uneven cache placement in [4] is followed by the delivery scheme proposed herein.

Fig. 3 investigates the effect of the attenuation parameter ξ. It can be seen that for very small attenuation levels, the Multiserver scheme outperforms others. This is because, compared with the Multiserver scheme, our proposed scheme sacrifices the global caching gain (as $\hat{t} \leq \frac{KM}{S}$) for achieving a better multicast rate (i.e., $R_w \geq R_u$). However, when the attenuation level ξ is small, the difference between R_w and R_u is small, and hence, the rate improvement cannot compensate for the performance loss due to a smaller caching gain (i.e., $\frac{\hat{t}+L}{t+L} < R_u/R_w$ and hence $R_w^s < R_u^s$). Nevertheless, the proposed scheme outperforms all other schemes as the attenuation grows large. Note that the proposed memory allocation in (4) drastically improves the performance compared with the one proposed in [4] due to higher achieved DoF.

Figure 4 compares the performance for different user count (K) values. As depicted, the performance gap between the proposed method and the other schemes increases as K grows larger. This is because with larger K, the global caching gain $\hat{t}$ (hence, the symmetric rate) is improved. Finally, in Fig. 5, the cumulative distribution function of total delivery time is depicted. Note that without memory allocation process, bounded delivery time can not be guaranteed, making conventional CC approaches unsuitable for delay-constrained applications.

VI. Conclusion

We proposed a location-dependent multi-antenna coded caching scheme tailored for wireless immersive viewing applications. For the placement phase, we employed a memory allocation process to incentivize content relevant to wireless bottleneck areas, resulting in a non-uniform, location-dependent cache placement. Then, during the delivery phase, we used a novel codeword creation process to enable global caching and spatial multiplexing gains jointly, while serving each user with a dedicated rate. Numerical experiments demonstrated significant performance gains compared to the state-of-the-art.

References

[1] Cisco, "Cisco Annual Internet Report, 2018–2023," *White Paper*, vol. 1, march, 2020.

[2] N. Rajatheva, I. Atzeni, E. Bjornson, A. Bourdoux, S. Buzzi, J.-B. Dore, S. Erkucuk, M. Fuentes, K. Guan, Y. Hu *et al.*, "White paper on broadband connectivity in 6g," *arXiv preprint arXiv:2004.14247*, 2020.

[3] E. Bastug, M. Bennis, M. Médard, and M. Debbah, "Toward interconnected virtual reality: Opportunities, challenges, and enablers," *IEEE Communications Magazine*, vol. 55, no. 6, pp. 110–117, 2017.

[4] H. B. Mahmoodi, M. Salehi, and A. Tölli, "Non-symmetric coded caching for location-dependent content delivery," in *2021 IEEE International Symposium on Information Theory (ISIT)*, 2021, pp. 712–717.

[5] M. Chen, W. Saad, and C. Yin, "Virtual reality over wireless networks: Quality-of-service model and learning-based resource management," *IEEE Trans. on Comm.*, vol. 66, no. 11, pp. 5621–5635, 2018.

[6] C. Yang, Y. Yao, Z. Chen, and B. Xia, "Analysis on cache-enabled wireless heterogeneous networks," *IEEE Transactions on Wireless Communications*, vol. 15, no. 1, pp. 131–145, 2015.

[7] H. Liu, Z. Chen, and L. Qian, "The three primary colors of mobile systems," *IEEE Comm. Magazine*, vol. 54, no. 9, pp. 15–21, 2016.

[8] G. S. Paschos, G. Iosifidis, M. Tao, D. Towsley, and G. Caire, "The role of caching in future communication systems and networks," *IEEE Journal on Selected Areas in Communications*, vol. 36, no. 6, 2018.

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

[10] M. Salehi, K. Hooli, J. Hulkkonen, and A. Tolli, "Enhancing next-generation extended reality applications with coded caching," 2022. [Online]. Available: https://arxiv.org/abs/2202.06814

[11] 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.

[12] H. Zhao, A. Bazco-Nogueras, and P. Elia, "Resolving the worst-user bottleneck of coded caching: Exploiting finite file sizes," in *2020 IEEE Information Theory Workshop (ITW)*. IEEE, 2021, pp. 1–5.

[13] A. Destounis, A. Ghorbel, G. S. Paschos, and M. Kobayashi, "Adaptive Coded Caching for Fair Delivery over Fading Channels," *IEEE Transactions on Information Theory*, 2020.

[14] M. Salehi, A. Tolli, and S. P. Shariatpanahi, "Coded Caching with Uneven Channels: A Quality of Experience Approach," in *IEEE International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, 2020, pp. 1–5.

[15] A. Tang, S. Roy, and X. Wang, "Coded caching for wireless backhaul networks with unequal link rates," *IEEE Transactions on Communications*, vol. 66, no. 1, pp. 1–13, 2017.

[16] G. Kramer and S. Shamai, "Capacity for classes of broadcast channels with receiver side information," in *IEEE Inform. Theory Workshop*, 2007.

[17] Z. Chen, H. Liu, and W. Wang, "A novel decoding-and-forward scheme with joint modulation for two-way relay channel," *IEEE Communications Letters*, vol. 14, no. 12, pp. 1149–1151, 2010.

[18] B. Asadi, L. Ong, and S. J. Johnson, "Optimal coding schemes for the three-receiver awgn broadcast channel with receiver message side information," *IEEE Trans. on Inform. Theory*, 2015.

[19] M. Salehi, H. B. Mahmoodi, and A. Tölli, "A low-subpacketization high-performance mimo coded caching scheme," *arXiv preprint arXiv:2109.10008*, 2021.

[20] H. B. Mahmoodi, M. Salehi, and A. Tölli, "Asymmetric coded caching for multi-antenna location-dependent content delivery," *available at http://arxiv.org/abs/2201.11611*, 2022.

[21] A. Charnes and W. W. Cooper, "Programming with linear fractional functionals," *Naval Research logistics quarterly*, 1962.