Mobility-Aware Coded Storage and Delivery

Emre Ozfatura and Deniz Gündüz

***Abstract*—We consider a cache-enabled heterogeneous cellular network, where mobile users (MUs) connect to multiple cache-enabled small-cell base stations (SBSs) during a video downloading session. SBSs can deliver these requests using their local cache contents as well as by downloading them from a macro-cell base station (MBS), which has access to the file library. We introduce a novel mobility-aware content storage and delivery scheme, which jointly exploits coded storage at the SBSs and coded delivery from the MBS to reduce the backhaul load from the MBS to the SBSs. We show that the proposed scheme provides a significant reduction both in the backhaul load when the cache capacity is sufficiently large, and in the number of sub-files required. Overall, for practical scenarios, in which the number of sub-files that can be created is limited either by the size of the files, or by the protocol overhead, the proposed coded caching and delivery scheme decidedly outperforms state-of-the-art alternatives. Finally, we show that the benefits of the proposed scheme also extends to scenarios with non-uniform file popularities and arbitrary mobility patterns.**

***Index Terms*—Coded caching, coded storage, heterogeneous cellular networks, MDS codes, mobility.**

I. Introduction

Due to the popularity of on-demand video streaming services, such as YouTube and Netflix, video content dominates the Internet traffic. A promising solution to mitigate the excessive video traffic and to reduce the latency in video streaming is to store popular contents at the network edge. There are two prominent caching approaches to reduce the backhaul load in cellular networks; namely, *coded storage* and *coded delivery*. Coded storage is designed to provide a certain flexibility to users by receiving a file from multiple access points without worrying about overlapping bits. Maximum distance separable (MDS) or fountain codes can be used for this purpose, for example, in multi-access downlink scenarios, e.g., a static user downloading content from multiple small-cell base stations (SBSs) [1]–[4], mobile users (MUs) connecting to different SBSs sequentially to download content [5]–[9], or MUs utilizing device-to-device (D2D) communication opportunities [10]–[15]. Coded delivery, on the other hand, utilizes caches at MUs to serve multiple demands simultaneously, and reduce the amount of data that must be transmitted [16]–[30]. Coded delivery schemes consist of two phases. In the *placement phase*, files are divided into sub-files, and each user stores a certain subset of the sub-files. In the *delivery phase*, the server carefully constructs multicast messages as XORed combinations of requested sub-files. This multicast gain increases with the number of users; that is, the higher the number of users the lower the per-user delivery rate. However, the number of sub-files has to increase exponentially with the number of users to achieve the promised caching gain.

To remedy this limitation designs with *low subpacketizaton levels* have been sought for. In [31], the authors show that a particular family of bipartite graphs, namely Ruzsa-Szemerédi graphs, can be used to construct a coded caching scheme, in which the number of sub-files scales linearly with the number of users, K, when K is sufficiently large. However, large K requirement limits the practical applicability of this design. Alternatively, linear block codes are used in [32] to reduce the number of sub-files dramatically with a small increase in the delivery rate for any K. Placement delivery arrays (PDAs), whose columns and rows represent the users and sub-files, respectively, are introduced in [33], which seek a trade-off between the delivery rate and the number of sub-files. Different PDA designs, using bipartite graphs [34], hypergraph approach [35], and user grouping strategy [36] have been introduced.

A. Our contributions

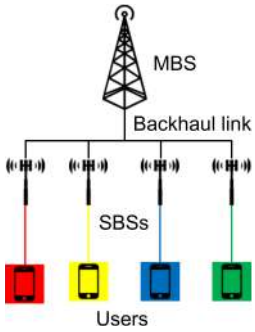
We consider a hierarchical network, in which MUs connect to cache-enabled SBSs to receive their requests. While a MU connects to a single SBS at any time, it may connect to multiple SBSs over the course of downloading its request. SBSs are connected to a macro-cell base station (MBS) through a shared backhaul link. We introduce a novel coded storage and delivery scheme, and show that the mobility of the users can be utilized to reduce the number of required sub-files. The proposed scheme divides the SBSs into smaller groups according to the mobility patterns of the MUs, and divides each file into equal-length fragments accordingly, so that MUs collect required fragments from different SBSs, and an independent coded delivery scheme is applied to each group of SBSs. Fragments are encoded by MDS-codes to guarantee that the MUs do not collect the same fragment more than once. We introduce an efficient grouping strategy for the SBSs via utilizing the analogous frequency reuse pattern problem [37]. Our contributions in this paper can be summarized as follows:

- We introduce a two-level coded storage and delivery scheme, and show that for a given MU mobility scenario, optimal solution for the proposed strategy can be analyzed as a cell coloring problem. We further provide the optimal coloring scheme that minimizes the delivery rate.
- We show that the proposed mobility-aware scheme can work as an add-on solution to improve the subpacketization-delivery rate trade-off, and can be combined with existing coded delivery designs with reduced subpacketization. We also show that even its implementation combined with the conventional coded delivery

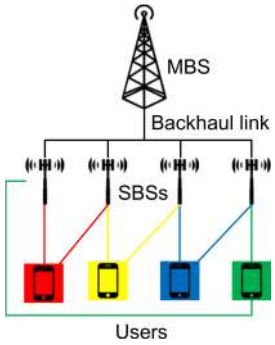
Emre Ozfatura and Deniz Gündüz are with Information Processing and Communications Lab, Department of Electrical and Electronic Engineering, Imperial College London Email: {m.ozfatura, d.gunduz} @imperial.ac.uk

This work was supported in part by the Marie Sklodowska-Curie Action SCAVENGE (grant agreement no. 675891), and by the European Research Council (ERC) Starting Grant BEACON (grant agreement no. 677854).

This paper was presented in part at the 2018 ITG Workshop on Smart Antennas in Bochum, Germany.



(a) The single-access model.



(b) The multi-access model with uniform access pattern.

Fig. 1: Illustration of the static user access models studied in [16] and [24], respectively.

scheme outperforms the state-of-the-art schemes specifically designed for reduced subpacketization.
- Finally, we show that the benefits of the proposed scheme extends also to non-uniform popularity distributions as well as to more general mobility scenarios.

B. Related Work

A closely related hierarchical network is considered in [24], in which a MBS serves multiple cache-equipped SBSs through a shared link, while each user is connected to L SBSs. It is shown that a lower delivery rate compared to [16] is achievable (see Fig. 1 for the models in [16] and [24]). Although it is not highlighted explicitly in [24], existence of a multi-access pattern reduces the number of required sub-files as well. The hierarchical network structure studied here also has similarities with *combination networks*, in which the MBS is connected to $\binom{k}{r}$ users through k relays. Coded delivery for combination networks is studied in [38] with caches at the users. The analysis is later extended to caches at the relay nodes [39], coded storage at the end-users [40], PDA design for reduced subpacketization [41], and combination networks with interference [42]. Although two-level code designs have been studied for combination networks, typically for symmetric connection patterns, such designs cannot be applied to the scenario with mobility studied here, since, unlike combination networks, where user-relay connections are known in advance, MU-SBS connections exhibit a time-varying behaviour, which is the main challenge addressed here.

Notations. Throughout the paper, for positive integer N, the set $\{1,\dots,N\}$ is denoted by $[N]$. We use $\bigoplus$ to denote the bit-wise XOR operation, while $\binom{j}{i}$ represents the binomial coefficient corresponding to the number of i-element subsets of a set with j elements. Finally, $\lceil\cdot\rceil$ and $\lfloor\cdot\rfloor$ represents *ceil* and *floor* operations, respectively.

II. System Model

A. Network model

We consider a cellular network architecture that consists of one MBS and K SBSs, $SBS_1,\dots,SBS_K$. The MBS has access to a content library of N files, $W_1,\dots,W_N$, each of size F bits. Each SBS is equipped with a cache memory of MF bits. The SBSs are connected to the MBS through a shared wireless backhaul link. We assume that there are Q MUs in the system, $U_1,\dots,U_Q$, which follow the *single-access model with mobility*; that is, each MU is connected to exactly one SBS at a particular time instant; however, due to mobility, it connects to multiple SBSs over time. We consider equal-length time slots, whose duration corresponds to the minimum time duration a MU remains connected to the same SBS. We also assume that each SBS is capable of transmitting B bits to a MU within one time slot. Hence, a file of size F bits can be downloaded in $T=\frac{F}{B}$ slots. We define the mobility path of a single MU as the sequence of small-cells visited during these T time slots. For instance, $T=3$, (SBS_2, SBS_3, SBS_4) is one such mobility path. The joint mobility pattern of the MUs is denoted by $V_T\in\mathcal{V}$, a $Q\times T$ random matrix whose ith row corresponds to the mobility path of the ith MU, while $\mathcal{V}$ denotes the set of all possible joint mobility patterns. The probability of a joint mobility pattern, $P(V_T)$, can be obtained from real data traces or by using a discrete time Markov process to model MU mobility behaviour.

The placement phase, during which the caches of the SBSs are filled, takes place before the demand vector $\mathbf{d}$ and joint mobility pattern V_T are known. Once the demands are revealed, where $d_i\in[N]$ denotes the demand of U_i, $i\in[Q]$, users are served by the SBSs they are connected to over the following T time slots. For a demand vector $d\triangleq[d_1,\dots,d_Q]$, the required delivery rate over the backhaul link, $R(M,N,K,\mathbf{d},V_T)$, is defined as the minimum number of bits that must be transmitted over the backhaul link during T time slots, normalized by the file size. The average delivery rate is given by

$$\bar{R}(M,N,K)=\mathbb{E}_{\mathbf{d},V_T}\left[R(M,N,K,\mathbf{d},V_T)\right], \tag{1}$$

where the expectation is over both the demand and mobility distributions.

We remark that it is difficult to provide a closed form expression for $\bar{R}(M,N,K)$ in general, hence, to establish a clear connection between the previously introduced schemes in [24] and [16], and to provide a fair comparison with the schemes proposed in [16] and [32], we will introduce our scheme under the following simplifying assumptions. First, we assume that all the files in the library are requested by the MUs with the same probability, i.e., W_n is requested by a MU with

probability $1/N$, $n \in [N]$. Under this assumption, if $N >> Q$ which is the case in realistic scenarios, it is safe to assume that all the MUs request a different content. For instance, when $Q = 30$ and $N = 10000$ the probability of each MU requesting a different file is 0.975. This assumption has been widely accepted in the coded caching literature as it also represents the worst case scenario. Under this assumption, the average delivery rate becomes independent of $\mathbf{d}$, hence we can remove the first expectation in (1). Second, we initially limit our attention to the *high mobility scenario*, which assumes exactly one MU in each cell at each time slot, i.e., $Q = K$, and MUs do not revisit the same cell within the same download session of T time slots. Under this assumption, as we later show, a coded caching scheme that achieves the same delivery rate for each possible joint mobility pattern V_T can be designed. Therefore, we can remove the second expectation in (1) as well, and obtain a tractable closed form expression for the average delivery rate. We remark that, even though the high mobility assumption is restrictive, once the coded caching scheme is designed, it can be used for any mobility scenario. Indeed, later in Section IV we will consider a general mobility scenario, modeled as a discrete time Markov process with $Q > K$ MUs and non-uniform demands.

To better motivate and explain our model and results, we will first explain the previously studied access models in the literature, and then provide a detailed explanation of the considered single-access model with mobility.

B. User access models

1) Static single-access model: In this model, each MU connects to exactly one SBS, as illustrated in Fig. 1a. This corresponds to the shared link problem introduced in [16]. The caching and coded delivery method introduced in [16] works as follows. For $t \triangleq \frac{MK}{N} \in \mathbb{Z}$, in the placement phase, all the files are cached at level t; that is, W_n, $n \in [N]$, is divided into $\binom{K}{t}$ non-overlapping sub-files of equal size, and each sub-file is cached by a distinct subset of t SBSs. Then, each sub-file can be identified by a subset $\mathcal{I}$, where $\mathcal{I} \subseteq [K]$ and $|\mathcal{I}| = t$, such that sub-file $W_{n,\mathcal{I}}$ is cached by SBS_k, $k \in \mathcal{I}$. In the delivery phase, for each subset $\mathcal{S} \subseteq [K]$, $|\mathcal{S}| = t+1$, all the requests of the SBSs in $\mathcal{S}$ can be served simultaneously by the MBS via multicasting $\bigoplus_{s\in\mathcal{S}} W_{d_s,\mathcal{S}\setminus\{s\}}$.

Thus, with a single multicast message the MBS can deliver $t+1$ sub-files, and achieve a *multicasting gain* of $t+1$. The achievable delivery rate is $R(M,N,K) = \frac{K-t}{t+1}$. We emphasize that the promised coded caching gain is obtained by dividing each file into $\binom{K}{t}$ sub-files, which grows exponentially with K. This limits the potential gain in practice for finite-size files.

The delivery rate for non-integer t values can be obtained as a linear combination of the delivery rates of the two nearest M values for which the corresponding t values are integers. This is achieved by *memory-sharing* between the caching and delivery schemes for those two M values. In the rest of the paper we will consider integer t values unless otherwise stated.

2) Static multi-access model: In this model, each user connects to multiple SBSs. A particular case of this problem is studied in [24], where each user connects to L SBSs following a certain cyclic pattern, where U_k connects to $SBS_k, \dots, SBS_{k+L-1 \mod K}$, $k \in [K]$. The case of $L = 2$ is illustrated in Figure 1b. In [24], the authors divide the SBSs into L groups, where the lth group consists of $\mathcal{G}_l \triangleq \{SBS_k : k \mod L = l\}$. In the placement phase each file is divided into L equal-size disjoint fragments, i.e., W_n^l is the lth fragment of file W_n. For each $l \in [L]$, all the fragments in $\mathcal{W}^l \triangleq \{W_1^l, \dots, W_N^l\}$ are cached by the SBSs in $\mathcal{G}_l$. For the placement of a particular group $\mathcal{G}_l$, we use the same caching scheme as in the static single-access model with $\hat{K} = K/L$ SBSs, each with a normalized cache size[1] of $\hat{M} = ML$. Therefore, each fragment of each file is cached at level $t \triangleq \frac{\hat{K}\hat{M}}{N} = MK/N$, i.e., sub-file $W^l_{n,\mathcal{I}}$, where $\mathcal{I} \subseteq \{k : k \mod L = l\}$ and $|\mathcal{I}| = t$, is cached by SBS_k, $k \in \mathcal{I}$. Similarly, the coded delivery phase, presented in Algorithm 1, is executed for each $\mathcal{G}_l$, $l \in [L]$, separately. The delivery rate is found as $R(M,N,K) = \frac{K-Lt}{t+1}$. We note that the delivery rate decreases with L, the number of SBSs each MU connects to. Moreover, the number of sub-files is $L\binom{K/L}{t}$, which provides a significant reduction in subpacketization.

Algorithm 1: Delivery phase of [24]

```
for l = 1 : L do
    for ĺ = 0 : L − 1 do
        for S ∈ {k : k mod L = l}, |S| = t do
            ⊕_{s∈S} W^l_{d_{(s−ĺ) mod K}, S\{s}}
        end
    end
end
```

C. Problem definition

Our aim in this section is to minimize the normalized delivery rate over the backhaul link under the single-access model with high mobility assumption for MUs. We also want to reduce the number of sub-files for a practical caching strategy. The single-access model with mobility can be treated similarly to the static single-access model: each file is divided into T disjoint fragments, each of which is considered as a separate file, so that the size of the library and the caches are scaled to NT and MT, respectively. The placement and delivery phases are as in [16], according to caching level $t = \frac{KMT}{NT} = KM/N$, and the delivery rate is $R(M,N,K) = \frac{K-t}{t+1}$, with $T\binom{K}{t}$ sub-files. Below we will present an alternative caching and delivery scheme that will reduce the number of required sub-files considerably.

The approach in [24] is not applicable when users do not follow uniform access patterns. MDS-coded caching can be employed when the users are mobile, or access the SBSs with non-uniform patterns [5], [6], [43]. The key advantage of MDS-coded caching at the SBSs is to reduce the amount of data that need to be cached at each SBS for each file.

Consider the simple example with $K = 4$ SBSs, where a MU connects to any 3 of them. Each file is divided into 3

[1]The cache capacity is normalized here with respect to the size of a fragment, which is $1/L$ of the original file.

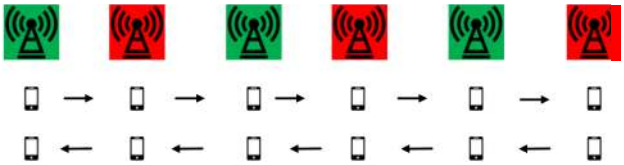


Fig. 2: Linear topology: moving along a line.

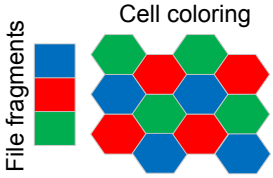


Fig. 3: File fragmentation and associated cell coloring.

fragments, which are encoded into 4 fragments with a (4,3) MDS code. Each SBS caches a different fragment so that a MU that connects to any three SBSs can recover the file. In this example, each SBS needs to cache only one fragment for each file, equivalently, $1/3$ of the original file. Under the high mobility assumption with $T = F/B$, it is sufficient to store only $1/T$ of each file at each SBS. Hence, if $M \geq N/T$, via MDS coded storage, the normalized delivery rate over the backhaul link can be reduced to zero, which means that all the MU requests can be delivered locally. The main drawback of MDS coded storage is that, coded delivery techniques cannot be applied directly to MDS coded files since the multicasting gain of coded delivery stems from the overlaps among cached sub-files at different SBSs. Hybrid designs which leverage both coded delivery and coded caching techniques have been previously studied for different network setups [39], [43], [44].

III. Solution Approach

In this section, we introduce a new hybrid design, which utilizes both MDS-coded caching and coded delivery to satisfy the demands of MUs under the single-access model with mobility, and analyze its performance under the high mobility assumption. For the sake of exposition, we first consider a special class of mobility patterns, for which the coded delivery technique in [24] can be applied directly.

A. Special case: Linear topology

Consider a linear mobility scenario, in which a MU's mobility path is determined by its direction and the first SBS it connects to (see Fig. 2). This can model, for example, MUs on a train connecting to SBSs located by the rail tracks in a known order.

Although a MU is connected only to the nearest SBS at any time instant, the coded delivery technique introduced in [24] for the static multi-access model can be applied in this special case. For given file size F and SBS transmission rate B bits per time slot, each file is divided into $T = F/B$ equal-size disjoint fragments, i.e., $W_n \triangleq (W_{n,1}, \ldots, W_{n,T})$, $n \in [N]$. Similarly, the set of all SBSs are also divided into T disjoint groups, denoted by $\mathcal{G}_1, \ldots, \mathcal{G}_T$, where SBSs in each group cache only one fragment of each file, using the placement scheme in [16]; that is, fragments $W_{n,l}$, $n \in [N]$, are cached across SBSs in group $\mathcal{G}_l$.

For a MU to be able to recover its request, the SBSs should be grouped such that any mobility path visits exactly one SBS from each group. Grouping of the SBSs can be considered as a coloring problem, where the SBSs are colored using T different colors such that any adjacent T of them have different colors. An example for $T = 2$ is illustrated in Fig. 2, where any two neighboring SBSs have different colors. Delivery phase is executed at each time slot separately for each group of SBSs $\{\mathcal{G}_l\}$, $l \in [T]$. Hence, for the special case of linear topology, the achievable delivery rate over the backhaul link is $R(M,N,K) = \frac{K-tT}{1+t}$ with $T\binom{K/T}{t}$ sub-files, where $t = KM/N$, and integer valued as before.

B. General Case: Two dimensional (2D) topologies

In this subsection, we consider a more general model in which the MUs move on a 2D grid, and each SBS covers a disjoint, equal-size area with hexagonal shape as illustrated in Fig. 3. As opposed to the linear model, it may not be possible to group all the SBSs using only T colors in the 2D model while ensuring that in any path of length T a MU connects to exactly one SBS from each group.

For given path length T, we will say that the SBSs are *L-colorable*, if there is a coloring of the SBSs with L colors such that any mobility path of length T consists of T SBSs with different colors. Note that we must have $L \geq T$. The following theorem states the achievable delivery rate over the backhaul link for an L-colorable network.

Theorem 1. *For given N, M, K, T, and $t \triangleq \frac{KMT}{NL}$, if the network is L-colorable, then the following delivery rate over the backhaul link is achievable using $T \times \binom{K/L}{t}$ sub-files, for integer t values:*

$$R(M,N,K) = \frac{K - tL}{1+t}. \tag{2}$$

Proof. In the placement phase, each file is divided into T disjoint equal-size fragments. These are then encoded into L fragments using an (L,T) MDS code. Hence, any T fragments out of the total L is sufficient to decode the original file. Consequently, each group of SBSs (SBSs with the same color) cache a different fragment using the placement scheme in [16]. The overall delivery phase consists of T identical consecutive delivery steps, each executed in one time slot, such that in each step a coded fragment is delivered to each MU, and having received T fragments at the end of T steps, each MU can recover the original file. To this end, we focus on a single delivery step. The number of SBSs in each group, labeled with the same color, is $\hat{K} = K/L$. If we consider the coded delivery phase for a particular group at a particular time slot, this is identical to the single-access model with $\hat{K}$ SBSs each with a cache memory of size $\hat{M} = MT$ files; with the corresponding delivery rate $\frac{\hat{K}-\hat{K}\hat{M}/N}{1+\hat{K}\hat{M}/N}\frac{1}{T} = \frac{K/L-t}{t+1}\frac{1}{T}$. Accordingly, the overall delivery rate is found as $R(M,N,K) = \frac{K-tL}{1+t}$. □

For non-integer t values the following lemma can be used to calculate the corresponding achievable delivery rate.

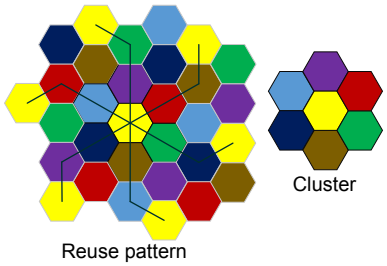


Fig. 4: Frequency reuse pattern with i=2, j=1 and the corresponding cluster.

Lemma 1. *If* $t = \frac{KMT}{NL}$ *is not an integer, then the following rate is achievable by* memory sharing

$$R(M,N,K) = \left(\gamma \frac{\frac{K}{L} - \lfloor t \rfloor}{\lfloor t \rfloor + 1} + (1-\gamma)\frac{\frac{K}{L} - \lceil t \rceil}{\lceil t \rceil + 1}\right) L, \qquad (3)$$

where $\gamma \triangleq \lceil t \rceil - t$.

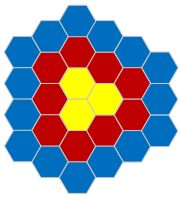
A simple example for $T = 2$ is illustrated in Fig. 3. As one can observe, three colors are sufficient to group the SBSs to ensure that in any mobility path of length two a MU always connects to two SBSs with different colors. Hence, in the placement phase of the given example, each file is initially divided into two fragments and these fragments are then encoded using (3,2) MDS code to obtain 3 coded fragments labeled with colors green, red, and blue. All the SBSs in the same group cache the fragments that have been assigned the same color. Then, at each time slot, the coded delivery phase is executed for each group of SBSs independently.

Remark 1. *In the single-access model with mobility, we can see from (2) that the rate increases with the number of colors. Hence, the goal is to identify the minimum L such that the network is L-colorable. In the next section, we study the optimal coloring strategy and the corresponding backhaul delivery rate.*

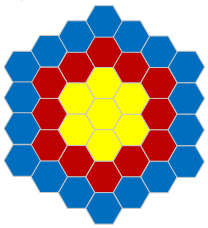
C. Cell coloring

For a given mobility path of length T, our objective is to color the cells with the minimum number of colors while ensuring that each color is encountered at most once in any mobility path. For given cell structure and mobility length T, the cell coloring problem can be modeled as a *vertex coloring problem* in a graph. Consider K SBSs with disjoint cells. We can consider each cell as a vertex of a graph G, and add an edge between vertices k and j if there is a mobility path of length T that contains both cell k and cell j. Once the graph G is constructed, its chromatic number $\gamma(G)$ gives the minimum L such that the network is L-colorable. However, this is an NP-hard problem [45]. Hence, finding the minimum L in a large network may not be feasible. Instead, we focus on the scaling behavior of L, i.e., for a given mobility path length T, how L scales as K goes to infinity.

We limit our focus on hexagonal cells first. This problem is analogous to the well-known *frequency reuse pattern* problem in cellular networks [37], where the same frequency



(a) Clusters for $T = 2$, $T = 4$ and $T = 6$ are illustrated with yellow, yellow and red, and all three colors, respectively.



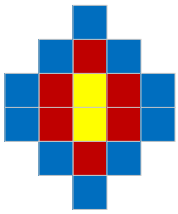
(b) Clusters for $T = 3$, $T = 5$ and $T = 7$ are illustrated with yellow, yellow and red, and all three colors, respectively.

Fig. 5: Examples of clusters for increasing T for hexagonel cells.

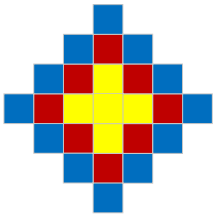
is allocated to multiple cells to efficiently use the limited spectrum while minimizing interference. Note that, frequency reuse patterns in the coded delivery framework has also been studied in [18] for limiting the interference in device-to-device communication. Cells serving in the same frequency are called *co-channel cells*. Co-channel cells are determined according to a constraint specified on the distance between the center of two co-channel cells. In [37], a frequency reuse pattern (or, equivalently, a co-channel cell pattern) is defined via the integer-valued shift parameters i and j: starting from a cell, "move i cells along any chain of hexagons; turn counter-clockwise 60 degrees; move j cells along the chain that lies on this new heading". An example with $i = 2$ and $j = 1$ is illustrated in Fig. 4. When co-channel cells are identified with the same color, the pattern of cells with different colors, which is repeated across the whole network, is called a *cluster*. An example of a cluster is illustrated in Fig. 4. It is shown that, using a reuse pattern with shift parameters i and j the two nearest co-channels are separated with a distance $D = \sqrt{3C}$ (scaled with the cell diameter), where $C = i^2 + j^2 + ij$ is the *cluster size*, which gives the total number of frequencies used.

We remark that when the nearest co-channel cells are separated with a distance $D = \sqrt{3C}$ according to the reuse pattern with shift parameters i and j, a MU in a particular cell should visit at least $i + j$ (including the current cell) cells to reach the nearest co-channel cell, and by definition no two of these cells can be co-channel cells. Therefore, the frequency reuse pattern problem is analogous to our problem, where the length of the mobility path T is equivalent to $i + j$, and the cluster size C is equivalent to the number of colors L. In our problem, we want to minimize the number of colors $L = T^2 - ij$ for a given mobility length $T = i + j$. Hence, we use the reuse pattern (i, j), with $i = \lceil \frac{T}{2} \rceil$ and $j = \lfloor \frac{T}{2} \rfloor$ to minimize the number of colors L. Examples of clusters with increasing T are shown in Fig. 5.

Remark 2. *When the reuse pattern* (i, j)*, with* $i = \lceil \frac{T}{2} \rceil$ *and* $j =$



(a) Clusters for $T = 2$, $T = 4$ and $T = 6$ are illustrated with yellow, yellow and red, and all three colors, respectively.



(b) Clusters for $T = 3$, $T = 5$ and $T = 7$ are illustrated with yellow, yellow and red, and all three colors, respectively.

Fig. 6: Examples of clusters for increasing T for square cells.

$\lfloor \frac{T}{2} \rfloor$ *is used, it is possible to reach a cell from any other cell in* T *steps in the corresponding cluster; that is, each cluster corresponds to a complete graph.*

Theorem 2. *For a network of SBSs with hexagonal cells and mobility path length* T*, the minimum* L *such that the network is* L*-colorable is given by*

$$L_{min} = \begin{cases} 3n^2, & \text{if } T = 2n, \\ 3n^2 + 3n + 1, & \text{if } T = 2n + 1, \end{cases} \tag{4}$$

for some positive integer n*.*

Proof. It is clear that L_{min} is achievable using the explained reuse pattern. Since the cluster corresponds to a complete graph, its chromatic number is equal to the cluster size L_{min}, which implies that it is not possible to color the network with less than L_{min} colors. □

This can be extended to other topologies. For instance, if we consider a square grid, the given reuse pattern can be modified by simply using 90-degree turns instead of 60.

Corollary 1. *For a network of SBSs with square cells and mobility paths of length* T*, the minimum* L *such that the network is* L*-colorable is given by*

$$L_{min} = \begin{cases} 2n^2, & \text{if } T = 2n, \\ 2n^2 + 2n + 1, & \text{if } T = 2n + 1, \end{cases} \tag{5}$$

for some positive integer n*.*

Scaling behavior of clusters in a square cell topology is illustrated in Fig. 6. We remark that the cluster size increases with the number of neighboring cells. For instance, for a mobility path of length $T = 4$, the cluster size is $L = 12$ for hexagonal cells whereas it is $L = 8$ for square cells. Recall that the delivery rate of the mobility-aware coded delivery scheme increases with L. Hence, the mobility-aware scheme performs better when there are fewer cells a MU can move into at each step, or equivalently, when the mobility pattern has less uncertainty.

IV. Numerical Results

We consider two network topologies with $K = 24$ and $K = 48$ SBSs of hexagonal shapes, respectively. We consider a mobility path of length $T = 2$ and assume, until Section IV.B, that MUs do not visit the same cell twice within T time slots. Hence, the cells are colored according to the reuse pattern $(i = 1, j = 1)$ with a total of $L = 3$ colors as in Fig. 3. We compare the performance of our mobility-aware coded delivery scheme with the Maddah-Ali-Niesen (MAN) scheme of [16], in terms of two metrics: the number of required sub-files and the normalized backhaul delivery rate. For each topology, we analyze the performance of these schemes for two different storage capacities of $M/N = 1/4$ and $M/N = 1/8$, respectively. The numerical results are presented in Table I.

For $K = 24$ we observe that our mobility-aware coded delivery scheme reduces the number of sub-files dramatically. For $M/N = 1/8$ the proposed scheme results in 12.5% increase in the delivery rate, while reducing the number of sub-files by approximately $1/72$. The more interesting results are observed with a larger cache size, i.e., $M/N = 1/4$. In this case, the proposed scheme outperforms the original coded delivery scheme in both performance metrics. At first glance this might be counterintuitive since there is a trade-off between the delivery rate and the number of sub-files [32]. However, the mobility-aware approach not only utilizes the multicasting gain, but also the multi-access gain. When $M/N = 1/4$, the number of required sub-files goes from 269000 down to 140 with the proposed scheme.

Next, we consider $K = 48$ SBSs. When $M/N = 1/8$ our scheme results in a 20% increase in the delivery rate, while reducing the number of sub-files by approximately four orders of magnitude. Hence, thanks to the proposed approach, coded delivery with caching can be implemented in practice with only 20% increase in the delay.

One can argue that the number of sub-files can also be reduced by simply clustering the SBSs to obtain two sub-networks with $K/2$ SBSs, and applying the coded delivery scheme to each sub-network independently. Indeed, the clustering approach could reduce the number of sub-files significantly; however, it leads to a further increase in the backhaul delivery rate. The results with the clustering approach, assuming two clusters, each consisting of $K/2 = 24$ SBSs, are included in Table I. When there are two clusters, the corresponding delivery rate is simply the sum of their delivery rates. Hence, the coded delivery scheme with two clusters uses the same number of sub-files as the coded delivery scheme for $K = 24$ SBSs, but twice the delivery rate. One can easily observe that for both $M/N = 1/8$ and $M/N = 1/4$ our mobility-aware coded delivery scheme outperforms the coded delivery scheme with two clusters in terms of both performance metrics. We also observe that the mobility-aware coded delivery approach becomes more efficient compared to the other two schemes, particularly for large storage capacity.

M/N	Coded delivery method and network scenario	Number of sub-files	Normalized delivery Rate
$\frac{1}{8}$	MAN [16], for $K = 24$	4048	5.25
	Mobility-aware coded delivery for $K = 24$	56	6
$\frac{1}{4}$	MAN [16], for $K = 24$	2.69×10^5	2.57
	Mobility-aware coded delivery for $K = 24$	140	2.4
$\frac{1}{8}$	MAN [16], for $K = 48$	2.45×10^7	6
	Coded delivery for $K = 48$ with clustering	4048	10.5
	Mobility-aware coded delivery for $K = 48$	3640	7.2
$\frac{1}{4}$	MAN [16], for $K = 48$	1.39×10^{11}	2.77
	Coded delivery for $K = 48$ with clustering	2.69×10^5	5.14
	Mobility-aware coded delivery for $K = 48$	2.57×10^4	2.66

TABLE I: Comparison of the proposed mobility-aware delivery scheme with the MAN scheme [16] and the coded delivery scheme with clustering.

Coded delivery method	Number of sub-files	Normalized delivery rate
MAN [16]	2.8×10^{12}	3.69
Mobility-aware coded delivery	251940	4
Coded delivery using (12,8) block code [32]	2.34×10^6	5.33
Coded delivery with two clusters	1.18×10^6	6.85

TABLE II: Comparison of the proposed scheme with the MAN scheme [16], the delivery scheme with clustering, and the coded delivery scheme of [32] that uses block code designs.

To highlight this, for $T = 2$, consider the extreme point $M/N = 1/2$. In this case the backhaul delivery rate is 0, while the number of sub-files is only 2.

We remark that a more sophisticated scheme, such as the one utilizing the erasure code design in [32], can be also applied to seek a balance between the number of sub-files and the delivery rate. To this end, we consider the scenario in Example 9 in [32], where there are $K = 60$ SBSs with hexagonal shapes and set $M/N = 1/5$ and $T = 2$. In this setup, the original coded delivery scheme achieves a slightly lower delivery rate compared to the mobility-aware scheme with approximately 10^7 times more sub-files. To illustrate the efficiency of the mobility-aware scheme we can limit the number of subfiles to be less than 10^7 in a practical implementation, and then compare the achievable delivery rates. Performances of the clustering method and the block code design in [32] under this subpacketization constraint are shown in Table II. One can observe that the proposed mobility-aware caching scheme outperforms both schemes.

A. Non-uniform file popularity

When the popularity of the files in the database is not uniform, performance of coded delivery schemes can be further improved by allocating more cache memory to popular files. In the case of non-uniform popularity, the objective is to minimize the expected delivery rate for a given popularity distribution. In the literature, several schemes have been introduced for the non-uniform demand problem [21]–[27]. A simple yet efficient scheme, proposed in [21], groups files according to their popularities. One particular implementation of this scheme is the *file removal strategy*, in which the whole library is divided into two groups, namely *popular* and *unpopular* files, and only the files in the first group are cached with equal cache allocation. Caching fewer more popular files decreases the delivery rate for these files, but increases the likelihood of requesting an uncached file. Hence, the optimal strategy decides the number of files to be cached.

Let $\Pi(K, M, N, N_c)$ denote the cache placement policy, where $N_c \leq N$ denotes the number of cached files. Under a cache placement policy Π, the required delivery rate for a given demand vector $\mathbf{d}$ can be written as the sum of two rate functions corresponding to cached and uncached files:

$$R_{total}(\Pi, \mathbf{d}) = R_c(M, N_c, K, \mathbf{d}) + R_u(N - N_c, \mathbf{d}). \quad (6)$$

The delivery rate corresponding to uncached files, R_u, is simply equal to the number of requests for unchached files, while the delivery rate corresponding to cached files can be written as,

$$R_c(M, N_c, K, \mathbf{d}) = R(M, N_c, K) \quad (7)$$

$$- \sum_{\tau=1}^{T} \sum_{l=1}^{L} \left(\gamma \frac{\binom{N_u^{(l,\tau)}(\mathbf{d})}{\lfloor t \rfloor + 1}}{\binom{\hat{K}}{\lfloor t \rfloor}} + (1-\gamma) \frac{\binom{N_u^{(l,\tau)}(\mathbf{d})}{\lceil t \rceil + 1}}{\binom{\hat{K}}{\lceil t \rceil}} \right) \frac{1}{T}, \quad (8)$$

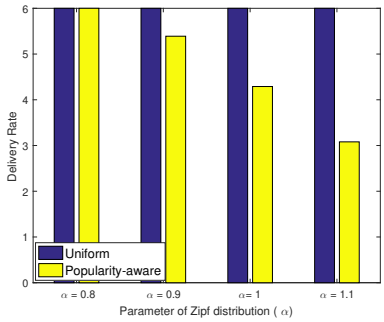
where $N_u^{(l,\tau)}$ denotes the number of uncached requests at time slot τ in cluster l, and $t \triangleq \frac{KMT}{N_c L}$. Let $\mathbf{P} = [p_1, \ldots, p_N]$ denote the popularity vector, where p_n is the probability of file W_n being requested. We assume, without loss of generality, that $p_1 \geq p_2 \geq \cdots \geq p_N$, and we define $p_c(N, N_c) \triangleq \sum_{n=1}^{N_c} p_n$ to denote the probability of requesting a cached file. Then, the expected total delivery rate, over the demand distribution, can be formulated as

$$\mathbb{E}_{\mathbf{d}}\left[R_{total}(\Pi, \mathbf{d})\right] = R(M, N_c, K) + \underbrace{\mathbb{E}_{\mathbf{d}}\left[\sum_{\tau=1}^{T} \sum_{l=1}^{L} N_u^{(l,\tau)}(\mathbf{d})\right] \frac{1}{T}}_{K(1-p_c(N,N_c))}$$

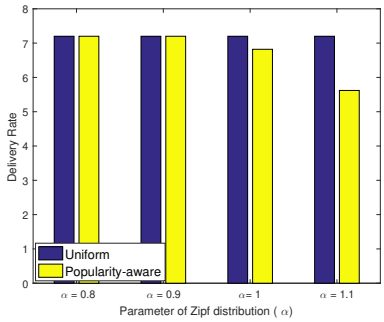
$$- \mathbb{E}_{\mathbf{d}}\left[\sum_{\tau=1}^{T} \sum_{l=1}^{L} \left(\gamma \frac{\binom{N_u^{(l,\tau)}(\mathbf{d})}{\lfloor t \rfloor + 1}}{\binom{\hat{K}}{\lfloor t \rfloor}} + (1-\gamma) \frac{\binom{N_u^{(l,\tau)}(\mathbf{d})}{\lceil t \rceil + 1}}{\binom{\hat{K}}{\lceil t \rceil}} \right)\right] \frac{1}{T} \quad (9)$$

$$\leq R(M, N_c, K) + K(1 - p_c(N, N_c)). \quad (10)$$

For given parameters K, M, N, L, T and $\mathbf{P}$, our goal is to find an optimal cache placement policy $\Pi^\star$ that minimizes



(a) M=150, N=1200 and K=24



(b) M=150, N=1200 and K=48

Fig. 7: Comparison between uniform and popularity-aware caching schemes.

(9). In general, the second term in (9) is negligible[2] compared to the rest; besides, it has no closed form expression, which makes it difficult to find the optimal policy $\Pi^{\star}$. Hence, for the popularity-aware cache placement strategy, we minimize the upper bound in (10) instead of (9). We can minimize (10) by searching over all possible values of N_c, with a computational complexity of $O(N)$.

To analyze the performance of the popularity and mobility-aware caching scheme, we consider the following setup with $M = 150$, $N = 1200$, $T = 2$, and $L = 3$. We assume that the popularity of the files follows a Zipf distribution with parameter α, which is the skewness of the distribution. We consider two scenarios with $K = 24$ and $K = 48$, and for each scenario we consider four different α values $\{0.8, 0.9, 1, 1.1\}$. The expected delivery rate for the uniform cache placement and popularity-aware cache placement schemes are compared in Fig. 7. For $K = 24$, we observe up to 50% reduction in the expected delivery rate by employing mobility-aware scheme together with popularity-aware placement compared to the mobility-aware scheme with uniform cache placement. We also observe that the impact of popularity-aware placement on the delivery rate is less visible when $K = 48$. We can conclude that popularity-aware placement has more significant impact for small K values, i.e., when t is small.

[2]The ratio between the first two term scales as $(K/N_u)^{t+1}$, thus when $t \geq 2$ the first term dominates the second term unless $K/N_u < 2$, but when $K/N_u < 2$, then R_u becomes the dominant factor.

B. Multi-user scenario with general mobility

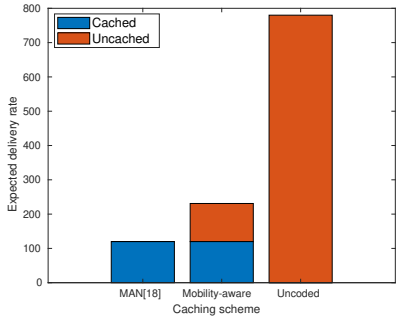
We have so far focused on the restrictive assumptions on the mobility and assumed that there is exactly one MU connecting to each SBS, and each MU visits a SBS with a particular color at most once. Now we will explain how the proposed mobility-aware scheme can be applied without any restrictions on mobility.

When we allow arbitrary mobility patterns, many users can end up in the same cell within the same time slot. However, in practice, SBSs, particularly those with small coverage areas, can serve only a limited number of MUs. We denote by Q_s the maximum number of MUs that can be served by a SBS in a time slot. We denote by λ the MU density, defined as the ratio between the average number of MUs connecting to a SBS and the service capacity Q_s. A higher λ means a busier network. At each time slot SBSs choose Q_s requests to serve and offload the remaining requests to MBS.

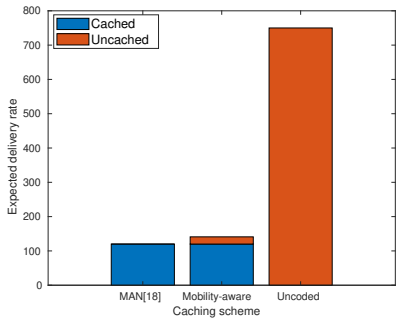
To account for users that may remain static, we employ a $(L+T, T)$ MDS code instead of an (L, T) MDS code for encoding fragments. The additional T encoded fragments are cached by the MBS. Hence, when a user remains connected to the same SBS, the SBS fetches from MBS this T additional coded fragments to serve the request. As in Section IV-A, we divide the MU requests into two: requests for the cached files and uncached files, respectively. Uncached file requests are observed either because the current cell is previously visited by the same MU, or requested file is not cached due to file removal strategy to allocate more cache size to popular files. Content delivery from the MBS to the SBSs is executed in two steps; first, requests for missing fragments of cached files are sent to the SBSs using coded delivery, then the remaining requests for the uncached files are fetched from the MBS via unicast transmission and served to the MUs. Since the service capacity of SBSs is limited by Q_s users, when there are more than Q_s requests, SBSs prioritize the requests for the cached files in order to reduce the delivery rate.

We will use a Markov model for mobility. Markov models have been shown to accurately represent real mobility traces, particularly for vehicular networks [46], [47]. In particular, we model MUs' mobility pattern with a discrete time Markov process, whose state is the current cell location. To analyze the impact of static MUs on the performance of the mobility-aware scheme, we assume that, at each time slot each MU stays connected to the current SBS with probability P_0, or move to one of the neighboring cells with equal probability.

For the simulations, we consider a 6×8 square grid topology with $K = 48$ SBSs, set $Q_s = 20$ and $T = 4$. As in the previous simulations, we set $M = 150$, $N = 1200$, and consider both uniform popularity and Zipf distribution with parameter $\alpha = 0.8$. For each case we consider both $P_0 = 0.2$ and $P_0 = 0.1$, modeling different mobility patterns. Therefore, we analyze 4 different scenarios in total, and for each scenario we further study two sub-scenarios by considering different MU densities $\lambda = 1.25$ and $\lambda = 1.75$.

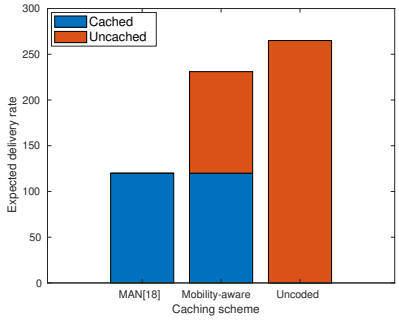


(a) $Q_s = 20, \lambda = 1.25$

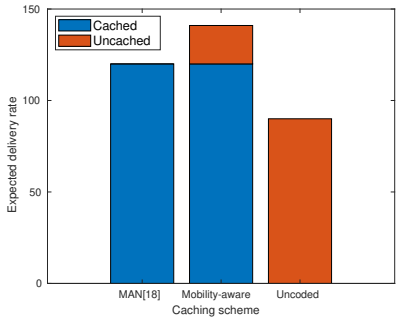


(b) $Q_s = 20, \lambda = 1.75$

Fig. 8: Comparison of the MAN, uncoded caching and mobility-aware caching schemes with $P_0 = 0.2$ and uniform content popularity.



(a) $Q_s = 20, \lambda = 1.25$



(b) $Q_s = 20, \lambda = 1.75$

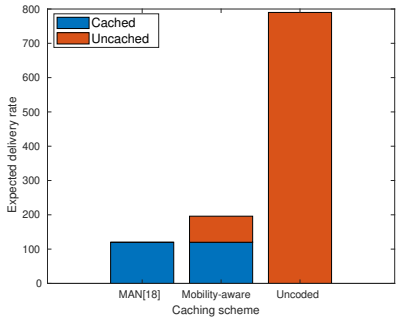
Fig. 9: Comparison of the MAN, uncoded caching and mobility-aware caching schemes with $P_0 = 0.2$ and content popularity with Zipf distribution.

We compare the MAN, uncoded caching[3] and the proposed mobility-aware caching schemes in terms of the expected total delivery rate $\bar{R}_{total}$. We note that under a general mobility pattern, there might be less than Q_s MUs in certain cells, or the number of requests for cached files might be less than Q_s. Nevertheless, to compute the coded delivery rate, we assume that there are Q_s requests for the cached files at each cell, so that the coded delivery scheme is executed Q_s times sequentially, and $R_C = Q_s \times R(M, N_c, K)$, where $N_c \leq N$ is the number of cached files. This serves as an upper-bound[4] on the coded delivery rate. The results (averaged over 1000 experiments) are illustrated in Figs. 8-11.

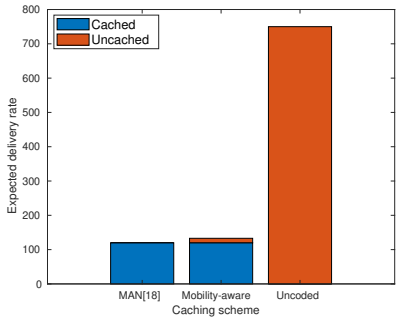
We observe from Fig. 8 that when MU density is low ($\lambda = 1.25$) the average total delivery rate of the mobility-aware scheme is almost twice that of the MAN scheme [16]. Although, according to the simulations, the request for uncached files constitute only 10 percent of all the served requests, they require almost the same delivery rate with the requests for cached files since multicasting gain cannot be exploited. However, when the MU density increases, since there is flexibility in choosing the requests for the cached files, the impact of the uncached file requests on the total delivery rate becomes marginal. Hence, under dense MU deployment

[3]Plots for uncoded caching only show requests for uncached files since requests for cached files are directly served from the SBSs, without requiring any rate from the MBS to the SBSs.

[4]We use this upper-bound to simplify our analysis, but the bound is tight when Q_s is sufficiently large. Indeed, in all these experiments, the difference between the exact coded delivery rate and the upper-bound is less than 3%.



(a) $Q_s = 20, \lambda = 1.25$



(b) $Q_s = 20, \lambda = 1.75$

Fig. 10: Comparison of the MAN, uncoded caching and mobility-aware caching schemes with $P_0 = 0.1$ and uniform content popularity.

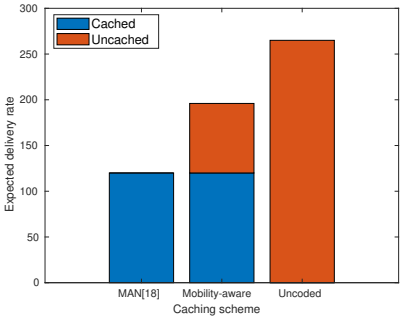
mobility-aware scheme becomes more efficient in terms of the expected delivery rate. We also want to highlight that the mobility-aware scheme requires only 80 sub-files while the MAN scheme requires almost 50 million sub-files, which is far from being practical.

In general, coded caching is expected to outperform uncoded caching approach, as we observe in Fig. 8. However, interestingly, results in Fig. 9 show that when the MU density is high and the file popularity has a skewed distribution, the uncoded caching scheme has a lower expected delivery rate. We observe similar performance trends in Figs. 10 and 11 as well. On the other hand, as expected, the performance of the mobility-aware scheme improves as P_0 decreases, which basically reduces the number of static MUs in the network. Specifically, for the mobility-aware scheme, we observe up to 35 percent reduction in the delivery rate due to requests for uncached contents, R_u, when P_0 reduces to 0.1 from 0.2. Finally, we also want to highlight that most of the requests for uncached files originate at the network boundaries, since those cells have fewer neighboring cells. Therefore, we expect the mobility-aware scheme to perform better when MU arrivals and departures to the network are also taken into account.

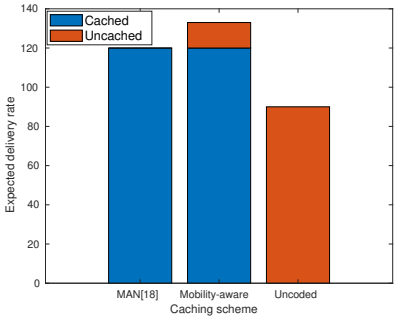
Simulation results indicate that in certain scenarios uncoded caching may outperform coded caching schemes. In the non-uniform demand scenario, a hybrid approach employing both uncoded and coded caching can be designed to reduce the delivery rate. The library can be divided into three groups based on popularity, and uncoded caching is used for the first group of most popular files, mobility-aware caching is used for the files in the second group, while the files in the third group are not cached at all. Given the MU mobility statistics (or, data traces), MU density statistics, and the file popularity statistics, an efficient file grouping strategy can be obtained to reduce the average delivery rate further. We will consider this hybrid approach in a future work.

V. Conclusions

We have introduced a novel MDS-coded storage and coded delivery scheme that adapts its caching strategy to the mobility patterns of the MUs. Our scheme exploits a coloring scheme for the SBSs, inspired by frequency reuse patterns in cellular networks, that have been extensively studied in the past for interference management. The files in the library are divided into sub-files, which are MDS-coded, and stored in the SBS caches, allowing MUs to satisfy their demands from multiple SBSs on their path under a high mobility assumption. We have shown that the proposed strategy achieves a significant reduction in the number of sub-files, which is critical in making caching algorithms practically relevant. Moreover, when the number of sub-files that can be created is limited, either due to the finite file size or to limit the complexity, the proposed scheme provides significant reduction in the backhaul load. We have also shown that the benefits of the proposed mobility-aware scheme extend also to non-uniform popularity distributions as well as to more general mobility scenarios allowing arbitrary random mobility patterns and multiple MUs being served by each SBS simultaneously.



(a) $Q_s = 20, \lambda = 1.25$



(b) $Q_s = 20, \lambda = 1.75$

Fig. 11: Comparison of the MAN, uncoded caching and mobility-aware caching schemes with $P_0 = 0.1$ and content popularity following Zipf distribution with parameter $\alpha = 0.8$.

References

[1] K. Shanmugam, N. Golrezaei, A. G. Dimakis, A. F. Molisch, and G. Caire, "Femtocaching: Wireless content delivery through distributed caching helpers," *IEEE Trans. Inf. Theory*, vol. 59, Dec. 2013.

[2] X. Xu and M. Tao, "Modeling, analysis, and optimization of coded caching in small-cell networks," *IEEE Trans. Commun.*, vol. 65, no. 8, pp. 3415–3428, Aug 2017.

[3] J. Liao, K. K. Wong, Y. Zhang, Z. Zheng, and K. Yang, "Coding, multicast, and cooperation for cache- enabled heterogeneous small cell networks," *IEEE Trans. Wireless Commun.*, vol. 16, no. 10, pp. 6838–6853, Oct 2017.

[4] S. Zhang, P. He, K. Suto, P. Yang, L. Zhao, and X. Shen, "Cooperative edge caching in user-centric clustered mobile networks," *IEEE Trans. Mobile Comput.*, vol. 17, no. 8, pp. 1791–1805, Aug 2018.

[5] K. Poularakis and L. Tassiulas, "Code, cache and deliver on the move: A novel caching paradigm in hyper-dense small-cell networks," *IEEE Trans. Mobile Comput.*, vol. 16, March 2017.

[6] E. Ozfatura and D. Gündüz, "Mobility and popularity-aware coded small-cell caching," *IEEE Commun. Lett.*, vol. 22, no. 2, pp. 288–291, Feb 2018.

[7] T. Liu, S. Zhou, and Z. Niu, "Mobility-aware coded-caching scheme for small cell network," in *2017 IEEE International Conference on Communications (ICC)*, May 2017, pp. 1–6.

[8] E. Ozfatura, T. Rarris, D. Gunduz, and O. Ercetin, "Delay-aware coded caching for mobile users," in *2018 IEEE 29th Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, Sep. 2018, pp. 1–5.

[9] E. Ozfatura and D. Gunduz, "Mobility-aware coded storage and delivery," in *Int'l ITG Workshop on Smart Antennas*, March 2018, pp. 1–6.

[10] M. Chen, Y. Hao, L. Hu, K. Huang, and V. K. N. Lau, "Green and mobility-aware caching in 5G networks," *IEEE Trans. Wireless Commun.*, vol. 16, no. 12, pp. 8347–8361, Dec 2017.

[11] T. Deng, G. Ahani, P. Fan, and D. Yuan, "Cost-optimal caching for d2d networks with user mobility: Modeling, analysis, and computational approaches," *IEEE Trans. Wireless Commun.*, vol. 17, no. 5, pp. 3082–3094, May 2018.

[12] J. Pedersen, A. Graell i Amat, I. Andriyanova, and F. Brännström, "Optimizing MDS coded caching in wireless networks with device-to-device communication," *IEEE Trans. Wireless Commun.*, vol. 18, no. 1, pp. 286–295, Jan 2019.

[13] ——, "Distributed storage in mobile wireless networks with device-to-device communication," *IEEE Trans. Commun.*, vol. 64, no. 11, pp. 4862–4878, Nov 2016.

[14] R. Wang, J. Zhang, S. H. Song, and K. B. Letaief, "Mobility-aware caching in d2d networks," *IEEE Trans. Wireless Commun.*, vol. 16, no. 8, pp. 5001–5015, Aug 2017.

[15] A. Piemontese and A. Graell i Amat, "MDS-coded distributed caching for low delay wireless content delivery," *IEEE Trans. Commun.*, vol. 67, no. 2, pp. 1600–1612, Feb 2019.

[16] M. A. Maddah-Ali and U. Niesen, "Fundamental limits of caching," *IEEE Trans. Inf. Theory*, vol. 60, no. 5, May 2014.

[17] ——, "Decentralized coded caching attains order-optimal memory-rate tradeoff," *IEEE/ACM Trans. Netw.*, vol. 23, no. 4, Aug 2015.

[18] M. Ji, G. Caire, and A. F. Molisch, "Fundamental limits of caching in wireless d2d networks," *IEEE Trans. Inf. Theory*, vol. 62, no. 2, pp. 849–869, Feb 2016.

[19] Q. Yu, M. A. Maddah-Ali, and A. S. Avestimehr, "The exact rate-memory tradeoff for caching with uncoded prefetching," *IEEE Trans. Inf. Theory*, vol. 64, no. 2, pp. 1281–1296, Feb 2018.

[20] M. M. Amiri and D. Gunduz, "Fundamental limits of coded caching: Improved delivery rate-cache capacity tradeoff," *IEEE Trans. Commun.*, vol. 65, no. 2, pp. 806–815, Feb 2017.

[21] U. Niesen and M. A. Maddah-Ali, "Coded caching with nonuniform demands," *IEEE Trans. Inf. Theory*, vol. 63, no. 2, Feb 2017.

[22] M. Ji, A. M. Tulino, J. Llorca, and G. Caire, "Order-optimal rate of caching and coded multicasting with random demands," *IEEE Trans. Inf. Theory*, vol. 63, no. 6, June 2017.

[23] J. Zhang, X. Lin, and X. Wang, "Coded caching under arbitrary popularity distributions," *IEEE Trans. Inf. Theory*, vol. 64, no. 1, pp. 349–366, Jan 2018.

[24] J.Hachem, N. Karamchandani, and S. N. Diggavi, "Coded caching for multi-level popularity and access," *IEEE Trans. Inf. Theory*, vol. 63, no. 5, May 2017.

[25] A. M. Daniel and W. Yu, "Optimization of heterogeneous coded caching," *CoRR*, vol. abs/1708.04322, 2017.

[26] A. Ramakrishnan, C. Westphal, and A. Markopoulou, "An efficient delivery scheme for coded caching," in *Proc. Int'l Teletraffic Congress*, 2015, pp. 46–54.

[27] S. Sahraei, P. Quinton, and M. Gastpar, "The optimal memory-rate trade-off for the non-uniform centralized caching problem with two files under uncoded placement," *IEEE Trans. Inf. Theory*, vol. 65, no. 12, pp. 7756–7770, Dec 2019.

[28] E. Ozfatura and D. Gunduz, "Uncoded caching and cross-level coded delivery for non-uniform file popularity," in *2018 IEEE International Conference on Communications (ICC)*, May 2018, pp. 1–6.

[29] 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.

[30] M. Mohammadi Amiri, Q. Yang, and D. Gündüz, "Decentralized caching and coded delivery with distinct cache capacities," *IEEE Trans. Commun.*, vol. 65, no. 11, pp. 4657–4669, Nov 2017.

[31] K. Shanmugam, A.Tulino, and A. Dimakis, "Coded caching with linear subpacketization is possible using Ruzsa-Szemerédi graphs," in *IEEE Int'l Symposium on Inform. Theory (ISIT)*, June 2017, pp. 1237–1241.

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

[33] Q. Yan, M. Cheng, X. Tang, and Q. Chen, "On the placement delivery array design for centralized coded caching scheme," *IEEE Trans. Inf. Theory*, vol. 63, no. 9, pp. 5821–5833, Sep. 2017.

[34] Q. Yan, X. Tang, Q. Chen, and M. Cheng, "Placement delivery array design through strong edge coloring of bipartite graphs," *IEEE Commun. Lett.*, vol. 22, no. 2, pp. 236–239, Feb 2018.

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

[36] M. Cheng, J. Jiang, Q. Wang, and Y. Yao, "A generalized grouping scheme in coded caching," *IEEE Trans. Commun.*, vol. 67, no. 5, pp. 3422–3430, May 2019.

[37] V. H. Donald, "Advanced mobile phone service: The cellular concept," *Bell System Technical Journal*, vol. 58, no. 1, pp. 15–41, Jan 1979.

[38] M. Ji, M. F. Wong, A. M. Tulino, J. Llorca, G. Caire, M. Effros, and M. Langberg, "On the fundamental limits of caching in combination networks," in *IEEE Int'l Workshop on Signal Proc. Advances in Wireless Communications (SPAWC)*, June 2015, pp. 695–699.

[39] A. A. Zewail and A. Yener, "Coded caching for combination networks with cache-aided relays," in *2017 IEEE International Symposium on Information Theory (ISIT)*, June 2017, pp. 2433–2437.

[40] K. Wan, M. Jit, P. Piantanida, and D. Tuninetti, "On the benefits of asymmetric coded cache placement in combination networks with end-user caches," in *IEEE Int'l Symposium on Information Theory (ISIT)*, June 2018, pp. 1550–1554.

[41] Q. Yan, M. Wigger, and S. Yang, "Placement delivery array design for combination networks with edge caching," in *2018 IEEE International Symposium on Information Theory (ISIT)*, June 2018, pp. 1555–1559.

[42] A. R. Elkordy, A. S. Motahari, M. Nafie, and D. Gunduz, "Cache-aided combination networks with interference," *IEEE Trans. on Wireless Communications*, vol. 19, no. 1, pp. 148–161, Jan 2020.

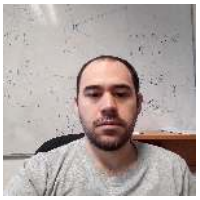
[43] N. Mital, D. Gunduz, and C. Ling, "Coded caching in a multi-server system with random topology," in *2018 IEEE Wireless Communications and Networking Conference (WCNC)*, April 2018, pp. 1–6.

[44] Y. P. Wei and S. Ulukus, "Novel decentralized coded caching through coded prefetching," in *2017 IEEE Information Theory Workshop (ITW)*, Nov 2017, pp. 1–5.

[45] D. Zuckerman, "Linear degree extractors and the inapproximability of max clique and chromatic number," in *Proc. ACM Symposium on Theory of Computing*, 2006, pp. 681–690.

[46] Z. Zhao, L. Guardalben, M. Karimzadeh, J. Silva, T. Braun, and S. Sargento, "Mobility prediction-assisted over-the-top edge prefetching for hierarchical vanets," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 8, pp. 1786–1801, Aug 2018.

[47] N. Abani, T. Braun, and M. Gerla, "Proactive caching with mobility prediction under uncertainty in information-centric networks," in *Proc. ACM Conf. on Information-Centric Networking*, 2017, pp. 88–97.



Emre Ozfatura He received his B.Sc. in Electronics Engineering with Math minor and M.Sc. in Electronics Engineering from Sabanci University (Turkey), in 2012 and 2015, respectively. He is currently pursuing his Ph.D. degree at Imperial College London, UK, where he is a member of the Information Processing and Communications (IPC) Lab. His research interests are video streaming applications, distributed content storage and distributed computation.

Deniz Gündüz [S'03-M'08-SM'13] received the M.S. and Ph.D. degrees in electrical engineering from NYU Tandon School of Engineering in 2004 and 2007, respectively. He is currently a Reader in information theory and communications at Imperial College London, UK. His research interests lie in the areas of communications and information theory, machine learning, and privacy. Dr. Gündüz is the Area Editor (for Machine Learning and Communications) for the IEEE Transactions on Communications, and serves as an Editor of the IEEE Transactions on Wireless Communications and IEEE Transactions on Green Communications and Networking. He is the recipient of the Best Paper Awards at the 2019 IEEE Global Conference on Signal and Information Processing (GlobalSIP) and the 2016 IEEE Wireless Communications and Networking Conference (WCNC), and the Best Student Paper Awards at the 2018 IEEE Wireless Communications and Networking Conference (WCNC) and the 2007 IEEE International Symposium on Information Theory (ISIT).