Mobility-Aware Coded Storage and Delivery

Emre Ozfatura and Deniz Gündüz

Abstract

Content caching at small-cell base stations (SBSs) is a promising method to mitigate the excessive backhaul load and delay, particularly for on-demand video streaming applications. A cache-enabled heterogeneous cellular network architecture is considered in this paper, where mobile users connect to multiple SBSs during a video downloading session, and the SBSs request files, or fragments of files, from the macro-cell base station (MBS) according to the user requests they receive. A novel content storage and delivery scheme that exploits coded storage and coded delivery jointly, is introduced to reduce the load on the backhaul link from the MBS to the SBSs. It is shown that the proposed caching scheme, by exploiting user mobility, provides a significant reduction in the number of sub-files required while also reducing the backhaul load when the cache capacity is large. Overall, for practical scenarios in which the number of subfiles that can be created is limited (by the size or the protocol overhead), the proposed coded caching and delivery schemes decidedly outperforms other known alternatives.

Index Terms

content delivery, coded storage, coded delivery, maximum distance separable (MDS) code, mobility, heterogeneous cellular networks, subpacketization, reuse patterns.

I. Introduction

Due to the popularity of on demand video streaming services, such as YouTube and Netflix, video dominates the Internet traffic [1], [2]. A promising solution to mitigate the excessive video traffic and to reduce the video latency in video streaming, particularly in cellular networks, is storing the popular contents at the network edge. There are two prominent approaches used extensively in the literature to reduce the backhaul load in cellular networks; namely, *coded storage* and *coded delivery*. In a broad sense, coded storage is designed from the perspective of the users, and allows them to efficiently receive a file from multiple access points without worrying about overlapping bits, which can be considered as "liquidification" of content. Maximum distance separable (MDS) or fountain codes, have been studied extensively, for example, in multi-access downlink scenarios, e.g., a static user downloading content from multiple small-cell base stations (SBSs) [3]–[6], mobile users (MUs) connecting to different SBSs sequentially to download content [7]–[11], or MUs utilizing device-to-device (D2D) communication opportunities [12], [13].

Coded delivery, on the other hand, is designed from the point of the server, which utilizes the caches of the users to seek multicasting opportunities in order to reduce the amount of data it needs to transmit to the users to satisfy their demands [14]–[24]. 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 the multicast messages as XORed combinations of the requested sub-files. Each user recovers its request from the multicasted messages together with its own cache contents. We note that the multicast gain increases with the number of users; that is, the higher the number of users the lower the per-user delivery rate. However, to achieve the promised gain, the number of sub-files has to increase exponentially with the number of users, which is considered as one of the main challenges in front of the implementation of coded delivery in practice [25].

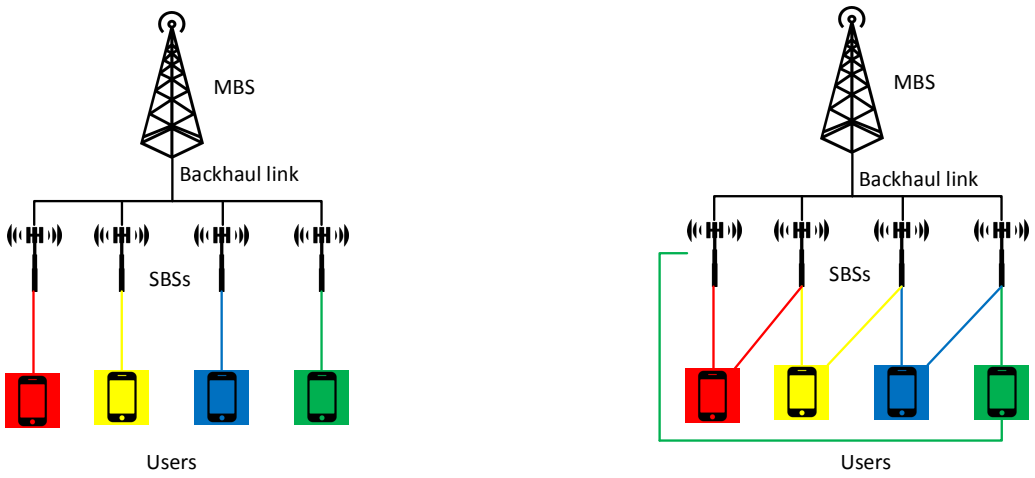
To remedy this limitation of coded delivery, coded caching and delivery designs that require *low subpacketizaton levels* have become an active research area. In [27], [28], the authors show that a particular family of bipartite graphs, namely Ruzsa-Szemerédi graphs, can be used to a construct a coded caching scheme, in which, for sufficiently large number of nodes, K, the number of sub-files scales linearly with K. Unfortunately, however, the large K assumption limits the practical applicability of the proposed coded caching design. Alternatively, in [29] a novel coded caching scheme based on linear block codes is introduced and it is shown that the number of sub-files can be reduced dramatically with a small increase in the delivery rate for any practical value of K. It is further shown that using different linear block codes, different delivery rates can be achieved using different sub-files for fixed K, M and N values, allowing some flexibility for implementation. In [30], coded caching designs based on linear block codes are constructed by using the so-called *placement delivery arrays (PDAs)*. We note that all the aforementioned works seek a coded caching design that reduces the number of sub-files with a minimum sacrifice in the delivery rate.

In this paper, we show that the mobility pattern of the users can be utilized to reduce the number of required sub-files by introducing a novel coded storage and delivery scheme, which is designed to take into account the random mobility patterns

Emre Ozfatura and Deniz Gündüz are withInformation 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. 725731).

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



(a) Single access model.

(b) Multi-access model with uniform access pattern.

Fig. 1: Illustration of the static user access models studied in [14] and [26], respectively.

of the users. The proposed scheme divides the SBSs into smaller groups according to the mobility patterns of the users, and applies coded delivery to each group of SBSs independently. MDS-coded storage is used to guarantee that the MUs can collect useful information from any of the SBSs they connect to until they recover their requested files. We introduce an efficient grouping strategy via utilizing the analogous well-known frequency reuse pattern problem [31].

In [26], a hierarchical network, in which a macro-cell base station (MBS) serves multiple cache-equipped SBSs through a shared link while each user is connected to L SBSs, is analyzed and it is shown that a lower delivery rate compared to [14] is achievable (please see Fig. 1 for illustration of the models considered in [14] and [26]). Although it is not highlighted explicitly in [26], existence of a multi-access pattern, i.e., each user accessing L SBSs, reduces the number of sub-files as well.

A similar observation is made in [32] by leveraging multiple transmission antennas instead of user mobility. Considering a MBS with L transmit antennas, the users are divided into groups of L, so that the channel from the MBS to each group becomes an $L \times L$ channel with L transmit antennas and L single-antenna users. For the delivery phase, a two-level coding scheme is used, where the outer code is designed considering each group of users as a single virtual user, and their L antennas as a single virtual antenna, and the coded caching scheme of [14] is applied for these L virtual users. Hence, each user in the same group caches the same sub-files. The inner code, for each group, is designed according to the $L \times L$ transmission channel from the server to the L users in this group.

The rest of the paper is organized as follows. The system model is introduced in Section II, and the proposed coded storage and delivery scheme is presented in Section III. The performance of the proposed scheme is evaluated and compared numerically with the coded delivery scheme in [14] in Section IV. Finally, in Section V we conclude the paper with a summary of our main contributions and potential future research directions.

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.

II. System Model

A. Network model

We consider a cellular network architecture that consists of one MBS and K SBSs, i.e., $SBS_1,\dots,SBS_K$. 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. Hence, when a user requests a content from a SBS, the SBS first checks its content cache. If the requested content is fully cached, then the SBS directly delivers the corresponding file. If the requested content is not cached at all, or partially cached, then the remaining parts of the content are first transferred from the MBS to the SBS over a backhaul link. We assume that all the demands from the MBS are delivered simultaneously over a shared error-free backhaul connection.

In this paper, we assume that all the files in the library are requested by the users 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 is large compared to K, which is the case in realistic scenarios, then it is safe to assume that all the users request a different content. For instance, when $K = 30$ and $N = 10000$ the probability of each user 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. We also assume that the number of users in the network

is limited by the number of SBSs; hence, there are K users in the network, i.e., $U_1, \dots, U_K$, and the requests of the users are denoted by the demand vector $\mathbf{d} \triangleq (d_1, \dots, d_K)$.

The placement phase, during which the caches of the SBSs are filled, takes place before the demand vector $\mathbf{d}$ is revealed. The required delivery rate for the backhaul link, $R(M, \mathbf{d})$, is defined as the minimum number of bits that must be transmitted over the shared link, normalized by the file size, for a given normalized cache capacity M, in order to satisfy all the user demands, $\mathbf{d}$. Since, we assume that each user requests a different file, we simply use $R(M)$ to denote the worst case delivery rate, instead of $R(M, \mathbf{d})$.

The delivery rate over the backhaul link from the MBS to the SBSs depends on the access model of the users. In this paper, we are particularly interested in the *single access model with mobility*, in which a MU is connected to exactly one SBS at a particular time instant; however, due to mobility, it connects to multiple SBSs over time. 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, it is assumed that each user is connected to exactly one SBS, as illustrated in Fig. 1a. This corresponds to the shared link problem introduced in [14]. The caching and coded delivery method introduced in [14] 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 MBS via multicasting

$$\bigoplus_{s \in \mathcal{S}} W_{d_s, \mathcal{S}\setminus\{s\}}. \tag{1}$$

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

The delivery rate for a cache memory M that results in a non-integer t value 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 [26], where each user connects to L SBSs following a certain cyclic pattern, where user 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 [26], the authors divide the SBSs into L groups, where the lth group consists of $\mathcal{G}_l \triangleq \{SBS_k : k \mod L = l\}$. Then, the coded delivery scheme in [14] is adapted to this setting as follows. 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. Then, 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 is executed for each $\mathcal{G}_l$, $l \in [L]$, separately. The coded delivery algorithm for this model is given in Algorithm 1, and the corresponding delivery rate is found as $R(M) = \frac{K-Lt}{t=1}$. We note that the delivery rate decreases with L, the number of SBSs each user connects to. The number of sub-files each file is divided into is $L\binom{K/L}{t}$ for this scheme, which provides a significant reduction in the subpacketization.

Algorithm 1: Delivery phase for the static multi-access model

```
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
```

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

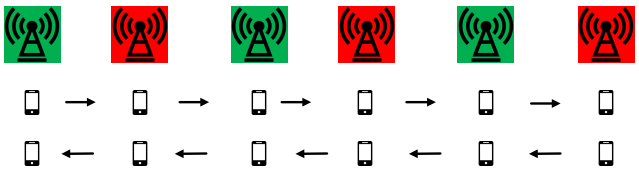


Fig. 2: Linear topology: moving along a line.

3) Single access model with mobility: In this model, MUs connect to different SBSs during the downloading period of a single-file, i.e., a video file or a group of frames, depending on the time scales. But, unlike in the previous model, each MU is connected only to the nearest SBS at any time instant. We consider equal-length time slots, whose duration corresponds to the minimum time duration a MU remains connected to the same SBS. We 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 user as the sequence of small-cells visited during these T time slots. For instance, for $K = 7$ and $T = 3$, SBS_2, SBS_3, SBS_4 is one such mobility path. We further assume that during the video downloading session of T time slots each MU is connected to exactly T different SBSs, which we call as the *high mobility assumption*.

C. Problem definition

Our aim is to minimize the normalized delivery rate over the backhaul link under the single access model with high mobility assumption for MUs. In addition to reducing the normalized backhaul delivery rate, we also want to reduce the number of sub-files used in the delivery phase in order to obtain a practically viable caching strategy.

We first note that the single access model with mobility can be treated similarly to the static single access model in the following way: each file is divided into T disjoint fragments, and each fragment is considered as a separate file so that the size of the file library and the size of the caches are scaled to NT and MT, respectively. Then the placement phase is executed as in [14] according to caching level $t = \frac{KMT}{NT} = KM/N$, and the delivery at each time slot can also be executed as in [14], and , $R(M) = \frac{K-t}{t+1}$ is still achievable with $L\binom{K}{t}$ sub-files. Below we will present an alternative caching and delivery scheme that will reduce the number of required sub-files considerably.

The approach introduced in [26] is an efficient method to reduce both the number of sub-files and the normalized delivery rate of the backhaul link, when the users connect to the SBSs in a uniform manner, as described in the previous section. However, this method is not applicable when the users do not follow the prescribed access patterns. Instead, MDS-coded caching can be employed when the users are mobile, or access the SBSs with non-uniform patterns [7], [8], [33]. 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 following simple example with $K = 4$ SBSs, where a MU can connect to any 3 of them. In this case, each file is divided into 3 fragments, and they are encoded into 4 fragments through 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. Accordingly, under the high mobility assumption with given $T = F/B$, it is sufficient to store only $1/T$ portion 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 user requests can be delivered locally. Otherwise, only MT/N portion of each file can be cached and delivered locally using MDS coded storage at the SBSs. 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 the cached sub-files at different SBSs. Hybrid designs which leverage coded delivery and coded caching techniques have been previously studied for different network setups [33]–[35].

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 [26] can be applied directly.

A. Special case: Linear topology

Consider a particular mobility scenario, in which a user's mobility path is determined by its direction and the first SBS it connects to. This can model, for example, MUs on a train connecting to SBSs located by the rail tracks in a known order. In this special case, MUs can be considered as moving on a line as illustrated in Fig. 2.

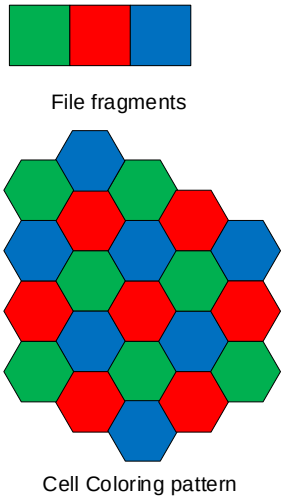


Fig. 3: File fragmentation and associated cell coloring.

Although a MU is connected only to the nearest SBS at any time instant, the coded delivery technique introduced in [26] for the static multi-access model can be applied in this special case. For given file size F and SBS transmission rate B, 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 [14]. That is, fragment $W_{n,l}$ are cached across SBSs in group $\mathcal{G}_l$.

For a MU to be able to recover all the sub-files of 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 for the backhaul link is $R(M) = \frac{K-tT}{1+t}$ with $T \times \binom{K/T}{t}$ sub-files, where $t = KM/N$, and integer valued as before.

B. General Case: Two dimensional topologies

In this subsection, we consider a general path model in which the users 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 one dimensional path model, in the 2 dimensional path model it may not be possible to group all the SBSs using only T colors 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 values of the variables 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:*

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

using $T \times \binom{K/L}{t}$ sub-files, for integer t values.

Proof. In the placement phase, each file is divided into T disjoint fragments with equal size. These are then encoded into L fragments using a (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 [14]. 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; and hence, the corresponding

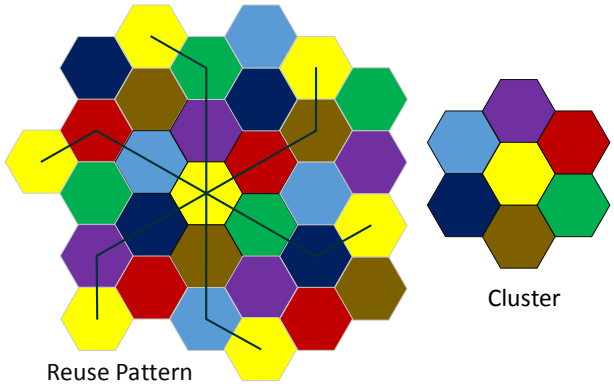


Fig. 4: Frequency reuse pattern with i=2, j=1 and the corresponding cluster.

delivery rate is $\frac{\hat{K}-\hat{K}\hat{M}/N}{1+\hat{K}\hat{M}/N}\frac{1}{T} = \frac{K/L-t}{t+1}\frac{1}{T}$, where $t \triangleq \frac{KMT}{NL}$. Accordingly, the overall delivery rate for the backhaul link is found as $R(M) = \frac{K-tL}{1+t}$. □

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

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

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

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

A simple example for $T = 2$ is illustrated in Fig. 3. As one can observe from Fig. 3, 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 which are labeled with colors green, red and blue. All the SBSs in the same group (i.e., those with the same color) 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. *We remark that the delivery rate achieved for a random path is higher than the one achieved for a deterministic path since the number of colors needed in general is greater than T, and it increases further with the number of colors used. Hence, in the single access model with mobility the objective is to identify the minimum L such that the network is L-colorable. In the following 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 using the minimum number of distinct colors while ensuring that, in any mobility path each color is encountered at most once. In general, for any given cell structure, and mobility length T, the cell coloring problem can be modeled as a vertex coloring problem. 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 j. Once the graph G is constructed, the chromatic number $\gamma(G)$ of this graph gives the minimum L such that the network is L-colorable. In general, this vertex coloring problem, i.e., finding the chromatic number $\gamma(G)$, is NP-complete. Hence, finding the optimal coloring, the minimum L, in a large network may not be feasible. Therefore in the scope of this paper we focus on the scaling behavior of the cell coloring problem i.e., for the given mobility path length T, what is the minimum L as K goes to infinity.

Let us limit our focus on hexagonal cells first. This problem is analogous to the well known *frequency reuse pattern* problem in cellular networks [31]. In this problem, the same frequency is allocated to multiple cells to efficiently use the limited available spectrum while minimizing interference. We note that utilization of frequency reuse patterns in the coded delivery framework has been previously studied in [16] for limiting the interference in device-to-device communication. The cells serving in the same frequency are called the *co-channel cells*. The co-channel cell locations are determined according to a given distance constraint (the distance between the center of two co-channel cells). In [31], a frequency reuse pattern (or, equivalently, a co-channel cell pattern) is defined via the integer-valued shift parameters i and j in the following way: 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". A frequency reuse pattern example with $i = 2$ and $j = 1$ is illustrated in Fig. 4. When co-channel cells are

(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: Scaling behavior of clusters for hexagonel cells.

identified with a same colour, then the pattern of cells with different colours, so that the whole network is a repetition of this pattern, is called the *cluster*, which 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 refers to the total number of different frequencies used in the network.

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 user 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. The scaling behavior of the clusters with respect to T is illustrated in Fig. 5.

Remark 2. *We observe that when the reuse pattern (i, j), with $i = \lceil \frac{T}{2} \rceil$ and $j = \lfloor \frac{T}{2} \rfloor$ is used for a given T, then in the corresponding cluster, it is possible to reach a cell from any other cell in T steps so that each cluster corresponds to complete graph.*

Theorem 2. *For a network of SBS with hexagonal cells, and a given 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 by using the proper reuse pattern explained. On the other hand, since the cluster corresponds to a complete graph, its chromatic index is equal to cluster size L_{min} which implies that it is not possible to color network with less than L_{min} different color. □

We remark that this analysis can be extended to different network 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. An example of a reuse pattern with i=2, j=2 is illustrated in Fig. 6 for a square grid.

Corollary 1. *For a network of SBS with square cells, and a given mobility path 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.

The scaling behavior of the clusters in a square cell topology is illustrated in Fig. 7. We remark that as the number of neighboring cells increases, the cluster size also increases. For instance, for a mobility path length $T = 4$, the corresponding cluster size is 12 for hexagonal cells whereas it is 8 for the square cells. Recall that the delivery rate of the mobility-aware coded delivery scheme increases with the cluster size L. Hence, we can conclude that the mobility-aware scheme performs better when there is less randomness in terms of the cells a MU can move into throughout its mobility path.

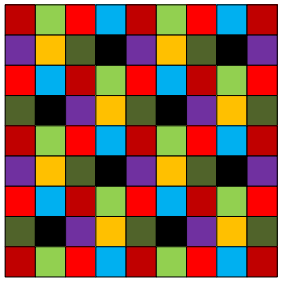
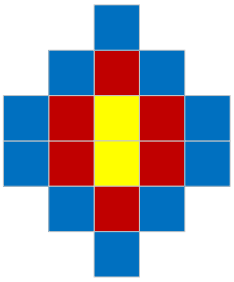
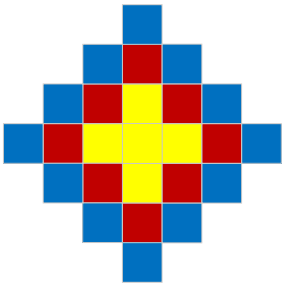


Fig. 6: Cell coloring in a square grid according to the reuse pattern with i=2, j=2.



(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. 7: Scaling behavior of clusters for square cells.

IV. Numerical Results

For the simulations, we consider two network topologies with $K = 24$ and $K = 48$ SBSs of hexagonal shapes, respectively. We consider a mobility path of length $T = 2$. 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 coded delivery scheme of [14] 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.

In the first group of simulations we analyze the network topology with $K = 24$ SBSs and observe that our mobility-aware coded delivery scheme reduces the number of sub-files dramatically. When, $M/N = 1/8$ our mobility-aware coded delivery scheme has 12.5% increase in the delivery rate, while reducing the number of sub-files by approximately $1/72$. The more interesting results are observed when the storage capability is higher, i.e., $M/N = 1/4$. In this case, the proposed mobility-aware coded delivery scheme outperforms the original coded delivery scheme in both performance metrics. At the first glance this might be counterintuitive since there is a trade-off between the delivery rate and the number of sub-files [29]. However, mobility-aware approach not only utilizes the multicasting gain, but also the multi-access, gain which is clearly visible in the deterministic path scenario. In this network setting the number of required sub-files goes from 269000 down to 140.

In the second group of simulations, we analyze the network topology with $K = 48$ SBSs. When $M/N = 1/8$ our mobility-aware coded delivery 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 practically realizable with only 20% increase in the delay.

At this point, one can argue that the number of sub-files could also be reduced by simply clustering the SBSs to obtain two sub-networks with $K/2$ SBSs, and then 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. We note that when there are two clusters, the corresponding delivery rate is simply the sum of the delivery rates corresponding to each cluster. Hence, the coded delivery scheme with two clusters uses the same number of sub-files as the coded delivery scheme for the network topology with $K = 24$ SBSs; however, the delivery rate is doubled. 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 when the storage capacity is high. To highlight this fact, for $T = 2$, consider the extreme point $M/N = 1/2$. In this case the backhaul delivery rate reduces to

Storage capacity (M/N)	Coded delivery method and network scenario	Number of sub-files	Normalized Delivery rate
$\frac{1}{8}$	Coded delivery [14], for $K = 24$	4048	5.25
	Mobility-aware coded delivery for $K = 24$	56	6
$\frac{1}{4}$	Coded delivery [14], for $K = 24$	2.69×10^5	2.57
	Mobility-aware coded delivery for $K = 24$	140	2.4
$\frac{1}{8}$	Coded delivery [14], 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
$\frac{1}{4}$	Coded delivery [14], 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 coded storage and delivery scheme with the conventional coded delivery scheme of [14] and a coded delivery scheme with clustering in terms of the number of required sub-files and the normalized delivery rate.

Coded delivery method	Number of sub-files	Normalized Delivery rate
Coded delivery [14]	2.8×10^{12}	3.69
Mobility-aware coded delivery	251940	4
Coded delivery using (12,8) block code [29]	2.34×10^6	5.33
Coded delivery with two clusters	1.18×10^6	6.85

TABLE II: Comparison of the proposed mobility-aware coded storage and delivery scheme with the conventional coded delivery scheme of [14] and a coded delivery scheme using block code designs terms of the number of required sub-files and the normalized delivery rate.

zero, while the number of subfiles is only two.

We remark that a more sophisticated scheme, such as the one utilizing the erasure code design in [29], 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 [29], where there are $K = 60$ SBSs with hexagonal shapes and $M/N = 1/5$. Similarly to the previous setup we consider $T = 2$. 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 files to be less than 10^7 and then compare the achievable delivery rates. Performances of the clustering method and the block code design in [29], under the subpacketization constraint are shown in Table II. One can observe that the proposed mobility-aware caching scheme outperforms the clustering scheme and the block code design both in terms of the delivery rate under the subpacketization constraint. At this point, it is worth emphasizing that, performance of the mobility-aware scheme depends on the mobility length T, thus to reduce the number of sub-files further, for each coded fragment placement scheme in [29] is used instead of the one in [14].

V. Conclusions

We have introduced a novel MDS-coded storage and coded delivery scheme that adopts its caching strategy to the mobility patterns of the users. 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 to reduce interference. The files in the library are divided into sub-files, which are MDS-coded, and stored in the SBS caches, allowing users 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. Particularly when the number of sub-files that can be created is limited, either due to the finite file size or to limit the complexity of the caching and delivery scheme, the proposed scheme provides significant gains in the backhaul load. We are currently extending our analysis to the case of non-uniform file popularity.

References

[1] Sandvine, "The Global internet phenomena report," October 2018, White Paper.

[2] Cisco, "Cisco visual networking : Forecast and methodology, 2017-2022," November 2018, White Paper.

[3] 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.

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

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

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

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

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

[9] 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.

[10] 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.

[11] E. Ozfatura and D. Gunduz, "Mobility-aware coded storage and delivery," in *WSA 2018; 22nd International ITG Workshop on Smart Antennas*, March 2018, pp. 1–6.

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

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

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

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

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

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

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

[19] 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.

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

[21] 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.

[22] A. Ramakrishnan, C. Westphal, and A. Markopoulou, "An efficient delivery scheme for coded caching," in *Proceedings of the 2015 27th International Teletraffic Congress*, ser. ITC '15. Washington, DC, USA: IEEE Computer Society, 2015, pp. 46–54.

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

[24] M. M. Amiri, Q. Yang, and D. Gunduz, "Coded caching for a large number of users," in *2016 IEEE Information Theory Workshop (ITW)*, Sep. 2016, pp. 171–175.

[25] G. Paschos, E. Bastug, I. Land, G. Caire, and M. Debbah, "Wireless caching: Technical misconceptions and business barriers," *IEEE Communications Magazine*, vol. 54, no. 8, pp. 16–22, August 2016.

[26] 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.

[27] K. Shanmugam, A. M. Tulino, and A. G. Dimakis, "Coded caching with linear subpacketization is possible using ruzsa-szemerédi graphs," in *2017 IEEE International Symposium on Information Theory (ISIT)*, June 2017, pp. 1237–1241.

[28] K. Shanmugam, A. G. Dimakis, J. Llorca, and A. M. Tulino, "A unified ruzsa-szemerédi framework for finite-length coded caching," in *2017 51st Asilomar Conference on Signals, Systems, and Computers*, Oct 2017, pp. 631–635.

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

[30] M. Cheng, J. Jiang, and Y. Yao, "A novel recursive construction for coded caching schemes," *CoRR*, vol. abs/1712.09090, 2017. [Online]. Available: http://arxiv.org/abs/1712.09090

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

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

[33] 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.

[34] 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.

[35] 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.