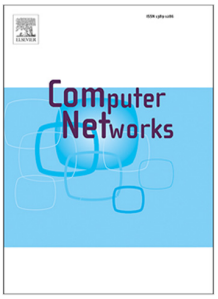
Journal Pre-proof



Content caching for shared medium networks under heterogeneous users' behaviours

Abdollah Ghaffari Sheshjavani, Ahmad Khonsari, Seyed Pooya Shariatpanahi, Masoumeh Moradian

PII: S1389-1286(21)00411-4
DOI: https://doi.org/10.1016/j.comnet.2021.108454
Reference: COMPNW 108454

To appear in: *Computer Networks*

Received date : 25 March 2021
Revised date : 21 June 2021
Accepted date : 2 September 2021

Please cite this article as: A.G. Sheshjavani, A. Khonsari, S.P. Shariatpanahi et al., Content caching for shared medium networks under heterogeneous users' behaviours, *Computer Networks* (2021), doi: https://doi.org/10.1016/j.comnet.2021.108454.

Content Caching for Shared Medium Networks Under Heterogeneous Users' Behaviours

Abdollah Ghaffari Sheshjavani*, Ahmad Khonsari*†, Seyed Pooya Shariatpanahi*, and Masoumeh Moradian†

Abstract—Content caching is a widely studied technique aimed to reduce the network load imposed by data transmission during peak time while ensuring users' quality of experience. It has been shown that when there is a common link between caches and the server, delivering contents via the coded caching scheme can significantly improve performance over conventional caching. However, finding the optimal content placement is a challenge in the case of heterogeneous users' behaviours. In this paper we consider heterogeneous number of demands and non-uniform content popularity distribution in the case of homogeneous and heterogeneous user-preferences. We propose a hybrid coded-uncoded caching scheme to trade-off between popularity and diversity. We derive explicit closed-form expressions of the server load for the proposed hybrid scheme and formulate the corresponding optimization problem. Results show that the proposed hybrid caching scheme can reduce the server load significantly and outperforms the baseline pure coded and pure uncoded and previous works in the literature for both homogeneous and heterogeneous user preferences.

Index Terms—Cache-aided communication, small cell networks, coded caching, heterogeneous user preference.

1 Introduction

1.1 Background

The global increase in the penetration of high throughput wireless devices such as tablets and smartphones has significantly facilitated the growing demand for mobile content through wireless media in recent years. Deploying small base stations (SBSs) to increase spatial reuse of the frequency spectrum by shrinking the network cells size is a promising solution to alleviate this growth and has stimulated many research initiatives [2]. Nonetheless, the high cost of wired links and the bottleneck of wireless links still poses as the main obstacle to providing high-speed backhaul to connect SBSs to the core network in this approach. To address this problem in content delivery scenarios, caching popular contents at these SBSs has been proposed to relieve the need for high-speed backhaul links [3]–[5].

*School of Electrical and Computer Engineering, College of Engineering, University of Tehran, Iran
†School of Computer Science, Institute for Research in Fundamental Sciences (IPM), Iran
Emails: {abdollah.ghaffari, a_khonsari, p.shariatpanahi}@ut.ac.ir, mmoradian@ipm.ir

. Some parts of this paper is the extended version of the problem that is presented at the WCNC 2020 conference [1].

In general, conventional caching methods attempt to cache the most popular contents located at close proximity of end-users such that the requests for popular contents are directly served from the local caches. This results in the so-called local caching gain, which is proportional to the local memory size. In [6], the authors introduced a novel coded caching scheme that significantly improves performance over conventional caching by leveraging the multicasting nature of the shared (such as wireless) medium even for caches with distinct demands. Their scheme, in addition to the local caching gain, results in global caching gain through using coded-multicast opportunities. The global caching gain is proportional to the aggregate memory of all the caches, where every user benefits from its cache contents to decode the desired content and to remove the interference in the coded message due to other caches requests. This idea has been further generalized to hierarchical coded caching [7], multi-server coded caching [8], decentralized coded caching [9]–[11], online coded caching [12], device to device (D2D) coded caching [13], hybrid server-D2D coded caching [14], coded caching with asynchronous user requests [15], and coded caching with multiple file requests [16]–[18].

To increase multicasting opportunities in coded caching, diverse parts of the library should be cached among different users, i.e. the *diversity principle*. However, in a set-up with non-uniform content popularity distribution, it is desirable to cache more popular contents with higher frequency, i.e. the *popularity principle*, which makes the cache contents of different users almost the same. As these two principles, namely diversity and popularity, are in tension, cache placement design in such scenarios is very challenging.

Heterogeneity can affect on the tension between diversity and popularity by affecting on multicasting opportunities. We can divide heterogeneity in caching into two main architectural and users' behavioral categories. Architectural heterogeneity includes different content and cache sizes and unequal users' downloading rate [19]–[22]. Users' behavioral heterogeneity includes different users' preferences [23], [24] and different numbers of users' requests at each time slot [25].

In this paper, we investigate the content caching in a shared medium network and propose a hybrid coded-uncoded caching under heterogeneous users' behaviors in order to minimize the shared medium traffic volume. we consider a caching scheme which partitions the contents

into three groups; coded-cached, uncoded-cached, and non-cached ones. In particular, we first focus on the problem of coded caching in a scenario with non-uniform user-independent (homogeneous-i.e., it is identical for all users) content popularity distribution and multiple demands, where different caches, which are connected directly to the server, may request different numbers of contents in each query. Then, we generalize this problem to non-uniform user-dependent (heterogeneous) content popularity distribution and non-equal cache sizes. In both cases, we derive explicit closed-form expressions of the shared medium traffic for the proposed hybrid coded caching scheme. In fact, this scheme proposes the optimal trade-off between popularity (uncoded caching gain) and diversity (coded caching gain).

In practice, the proposed heterogeneous caching scenario corresponds to a cellular network that includes a Macro Base Station (MBS) and multiple SBSs where each SBS is equipped with a limited size cache and serves multiple users. In this regard, on one hand, the number of requests each SBS sends to MBS in each query (or time slot), depends on the number of users it serves. On the other hand, the number of requests for each content depends on the popularity distribution over the SBS coverage area.

Finally, the numerical and simulation results show that the proposed hybrid caching outperforms the baseline pure coded, pure uncoded and previous works as well as the two-partitioning scheme reported in [26]–[30] for both SBS-independent and SBS-dependent content popularity distributions.

1.2 Organization

The rest of the paper is organized as follows. In Section 2 an overview of the coded caching is provided and related works are reviewed. In Section 3, the system model is introduced. The proposed caching schemes for multiple requests with SBS-independent and SBS-dependent non-uniform demands is described in Section 4 and 5, respectively. This is followed by numerical analysis and simulation results in Section 6. Finally, Section 7 concludes the paper.

2 Background of Coded Caching and Related Works

In this section, we first summarize the coded caching scheme reported in [6] and then review the subsequent related works.

2.1 Background on Coded Caching

The authors in [6] consider a system with one server connected through a shared, error-free link to K users. The server access to the database of N contents each of size F bits. Each user is equipped with a cache memory of size MF bits. Their system operates in two phases: a *placement* phase and a *delivery* phase. In the placement phase, each content is split into $\binom{K}{T}$ non-overlapping equal-sized sub-files, where $T = K \times M/N$ and the size of each sub-file is equal to $F/\binom{K}{T}$. The sub-files are distributed at caches such that each cache stores M/N of each content. Moreover, each sub-file has T copies in T different caches. In the delivery phase, each cache receives a request for a single content. The server then XORs the required sub-files by different caches according to a specific coding strategy and multicasts coded messages to the corresponding groups of $T+1$ caches. The achievable rate of the coding strategy for serving all contents at the shared link is proven to be [6]:

$$R = K\left(1-\frac{M}{N}\right)\min\left\{\frac{1}{1+K\times M/N}, \frac{N}{K}\right\}. \quad (1)$$

Where $K(1-\frac{M}{N})$ is the local caching gain and $\frac{1}{1+K\times M/N}$ is the global caching gain.

2.2 Related Works

Although the original coded caching scheme introduced in [6] performs well under homogeneous systems, the scheme is inefficient in non-uniform and heterogeneous content popularity and also heterogeneous architectural scenarios [20]–[31]. In the non-uniform content popularity (user-independent), different contents have different popularity but the popularity of any particular content is the same for all users [25]–[31]. On the other side, in the user-dependent content popularity, in addition to the fact that the popularity of contents is not the same, also the popularity of each particular content is not the same for different users [23], [24]. User-dependent content popularity is also called user preference in the literature.

To handle coded caching for non-uniform content popularity scenarios, one major approach in the literature is grouping contents based-on their popularity [25]–[31]. For the first time, authors in [31] proposed a grouping method to address non-uniform content popularity. In their method, in the placement phase, the library is partitioned into almost equiprobable groups and each user's cache is evenly shared among these groups. Finally, each group is treated as a single coded caching problem originally proposed in [6]. The efforts in [26]–[30] show that the asymptotically optimum placement strategy of grouping method of [31] is to partition the library into two groups: the popular contents are cached according to the scheme in [6] while the non-popular contents are not cached at all. However, authors in [25] show by some examples that when each cache receives multiple requests, in the grouping method of [31], partitioning the library altogether into three groups improves caching performance over two-partitioning placement strategies. In their method, the first part is cached fully at all the cache memories, and the second part is cached according to the original coded caching paradigm and the last part is not cached at all. Nevertheless, the authors in [25] assume that the library is divided into multiple levels, based on varying degrees of popularity. Besides, this work does not consider the closed-form expressions for the optimum partitioning under arbitrary popularity distribution.

Some works, such as [32] show that for uniform popularity distribution, the caching strategy proposed in [6] can be improved by removing sending some redundancy in the delivery phase. For example, suppose we have one content A of size F bits, three users, and a caching size of $1/3\times F$ bits for each user. Based-on the placement strategy of [6], A is divided into 3 pieces A_1, A_2, A_3 which each piece cached in the corresponding user. In the delivery phase, the scheme in [6] suggests to broadcast $A_1 \oplus A_2$, $A_1 \oplus A_3$ and

$A_2 \oplus A_3$ that provide a delivery rate of 1. But $A_2 \oplus A_3$ can be recovered from $(A_1 \oplus A_2) \oplus (A_1 \oplus A_3)$, and therefore we do not need to broadcast it and the delivery rate reduces to $2/3$. Authors in [33] generalized this idea to propose a new coded caching strategy under uncoded placement to handle non-uniform demands. This strategy uses equal sub-packetization for all contents while allowing to allocate more cache to more popular content. However, they propose and analyze the delivery strategy only for the case of the existing two contents in the system.

Other approaches, such as [24], [34] used a structured clique cover algorithm for all demands of the users to handle coded caching with non-uniform content popularity and other aspects of heterogeneity. In [34], the authors introduced a coded caching scheme where the users have more than one request, and the content popularity is non-uniform. They used a random popularity-based algorithm for cache placement and adapting the idea of dividing each content into equal-sized sub-contents. Then, subsequently they used a greedy constrained local graph coloring technique to find multicast opportunities in the delivery phase. Authors in [24] consider heterogeneous user preferences. In this work, each user caches its most probable content at the placement phase, then in the delivery phase, based on the request and cached matrix, tries to gain from multicast opportunities. Therefore, these works do not consider any optimization for the placement phase.

The architectural heterogeneity in coded caching is considered in some works, such as [19]–[22]. Authors in [19] studied the coded caching with unequal link rates and proposed the use of nested coded modulation (NCM) coding in the delivery phase. However, the main drawback of this work is that the cache size for each user needs to be correctly allocated to adapt NCM transmission, in a way that users with lower link rate need a larger cache size. In [22], the authors analyzed the coded caching problem in a generalized scenario of the D2D coded caching [13], where the cache size of users is unequal. In addition, coded caching under non-uniform file-length, non-uniform users cache size, and non-uniform content popularity is considered in [20]. This work shows that finding optimal caching with the three aforementioned heterogeneity has exponential complexity. Therefore, they developed a tractable optimization problem corresponding to a caching scheme with the above three heterogeneities and showed numerically that it performs well compared to the original exponentially scaling problem. Authors in [21] also considered the coded caching problem under non-uniform users' cache size and download rate. Some works, such as [35], [36] considered heterogeneous quality-of-service requirements in which each content may have various resolution copy based on the different users' device resolution requirements. However, these works do not consider the heterogeneity of user behaviors, such as user-dependent content popularity, while, previous studies indicate that the global popularity cannot be directly used to infer the local popularity of contents [37].

Recently, a few efforts consider heterogeneous content popularity in coded caching. The authors in [23] considered some aspects of two categories of heterogeneity, such as heterogeneous content sizes, heterogeneous cache size, and user-dependent content popularity. Although this work considers full heterogeneous content popularity, it analyzes the problem only for a very small scale such as a two users / two files scenario. Authors in [38] also study a game-theoretic perspective of coded caching. The results of this work show that heterogeneous user preferences have a significant impact on caching gains.

TABLE 1: Summery of the main notation.

Symbol	Explanation
n	generic content
c	generic SBS (cache)
i	generic step of sending coded contents
N	number of contents
F	size of each content(bit)
K	number of SBS
M_c	cache capacity of SBS c (content)
Z_c	number of users (demands) in range of SBS c
Z_{max}	$\max(\{Z_c\}_{c=1}^{K})$
D_c	demand vector of SBS c
$q_{c,j}$	probability of requesting the jth distinct coded content at the next request in SBS c
$p_{n,c}$	popularity of content n in SBS c
q_c^{coded}	the queue of the coded requests of SBS c
$q^{uncoded}$	the queue corresponding to uncoded requests
$P_i^{(c)}$	probability of at least i distinct requests in q_c^{Coded} at step i
Q_i	number of non-empty coded queues at step i
l_c	number of distinct request in the q_c^{coded}
$Q_i^{(c)}$	number of non-empty coded queues in first c caches at step i
$l_c^{(z)}$	number of distinct coded request in first z requests that received by SBS c
g	generic group of SBSs
$Y_{n,c}$	indicate content n is (or not) cached in SBS c
$X_{n,g}$	indicate content n is (or not) cached in group g
$S_{c,g}$	indicate SBS c is (or not) participate in group g
r_1	expected traffic load (MBS) for the coded contents
$r_1^{(i)}$	r_1 at step i
r_2	expected traffic load (MBS) for un-cached contents
r	total expected traffic load of the MBS $(r_1 + r_2)$

3 System Model

We consider a cellular network that consists of one MBS, which is connected through a shared error-free link to K SBSs, as depicted in Fig. 1. The content library has $N = \{W_1, W_2, ..., W_N\}$ distinct contents that are all accessible by the MBS. Without loss of generality, we assume that all contents have the same size equal to F bits. Each SBS c has a cache memory of size $M_c \times F$ bits for some integer number of $M_c \in [0, N]$ and is responsible for serving Z_c users where $c \in \{1, 2, \ldots, K\}$. Each user can connect and receive data from the MBS and only one SBS.

Similar to previous works, our system operates in two phases: the content placement phase and the content delivery phase. The placement phase is carried out during off-peak times. In this phase, the caching strategy determines some functions of all contents $s_c = f_c(W_1, ..., W_N)$, $c \in \{1, 2, \ldots, K\}$ that must be cached at each SBS based on the system parameters such as cache memory constraint and content popularity, then the caches are filled with corresponding contents from the library.

In the delivery phase, only the MBS has access to the whole library. Also, in each delivery phase, users reveal their requests to the SBSs, and subsequently, the SBSs send the required requests to MBS. Moreover, each SBS receives one request from each connected user, resulting in a total of Z_c requests at each delivery phase. It is worth noting that due to the arbitrary number of users connected to SBS c (i.e., Z_c), the model is general, and the assumption of requesting one content by each user is not restrictive. Therefore, the number of requests that each SBS receives is also arbitrary, and each SBS may receive multiple requests for some contents of the library. Denote $D_c = [d_{1,c}, ..., d_{Z_c,c}]$ as the demand vector of SBS c, where $d_{i,c}$ is the content requested by user i. Moreover, the number of distinct contents in D_c can be between 1 and Z_c (due to the possibility of duplicate requests for contents by different users). Upon collecting the requests of the users in SBSs, the MBS receives the list of distinct requested contents from each SBS and then, sends the required files over the shared link to satisfy these requests.

The content popularity distribution is arbitrary and can be SBS-dependent. In addition, we assume that it does not change during the delivery phase. $p_{n,c}$ denote the probability of requesting the content W_n by users under coverage of SBS c where $n \in \{1, 2, \dots, N\}$, and $c \in \{1, 2, \dots, K\}$. Therefore, based on the content popularity distribution (i.e., $\{p_{n,c}\}_{n,c=1}^{N,K}$), the number of requests per content, and the demand vector of each SBS c (i.e., D_c) is a random vector. Consequently, the traffic load in the delivery phase, denoted by R, to satisfy all SBSs is also a random variable. Unlike the placement phase, in the delivery phase, the cost of network load is high since the available bandwidth of the shared link is the system's bottleneck, so we only consider the traffic load in the delivery phase.

In this paper, we first consider designing a content caching scheme for the case of SBS-independent content popularities in Section 4, as the assumption of homogeneous popularities has been adopted by many previous works. Then, we generalize the same problem to the case of SBS-dependent popularities and non-equal cache sizes in Section 5.

4 SBS-INDEPENDENT NON-UNIFORM CONTENT POPULARITY

In this section, we assume that the content popularity distribution is the same for all SBSs. Regarding that SBSs have almost the same hardware, in this section, we assume that the cache sizes are the same for all SBSs, as many previous works [6]–[10], [25]–[31]. Therefore $p_{n,c} = p_n$, and $M_c = M$ where $c \in \{1, 2, \dots, K\}$.

Whenever caches have multiple and different requests, applying pure coded caching increases the number of multicast transmissions. On the other hand, if some of the contents are cached in the caches completely in uncoded form, the number of multicast transmissions decreases at the expense of decreasing the global caching gain. Thus, there exists a trade-off between caching the contents in uncoded and coded forms. In this regard, in this section, we extend our previous grouping method of caching under non-uniform demands in [1] to non-uniform multiple requests and formulate the optimization problem for minimizing the load of the shared link by presenting the hybrid coded-uncoded caching scheme.

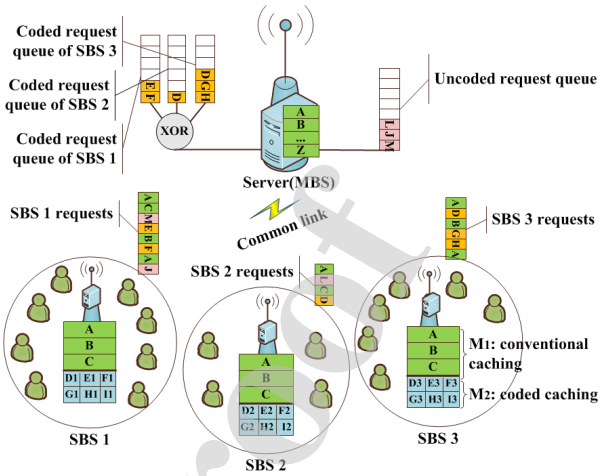


Fig. 1: An example of our caching system with one MBS containing $N = 26$ contents of size F bits connected via an error-free shared link to $K = 3$ SBSs, each with a cache size of $M_c \times F = 5F$ bits and serving Z_c end-users.

4.1 The proposed SBS-independent caching scheme

In the placement phase of the proposed caching scheme, we categorize the contents into (at most) three groups based on their request probabilities. In particular, we first choose the N_1 most popular contents among all N contents and then cache M_1 most popular contents among these N_1 selected ones entirely at all caches. Then, the remaining $N_1 - M_1$ contents are cached using the coded caching scheme proposed in [6]. Accordingly, each cache memory is divided into two parts: $M_1 \times F$ bits of each cache are allocated to the M_1 most popular contents, while the remaining $(M - M_1) \times F$ bits of memory are allocated to the coded caching scheme with a library size of $N_1 - M_1$ contents. In summary, the three groups of contents resulting from our scheme are:

1) M_1 most popular contents that are cached completely.
2) $(N_1 - M_1)$ popular contents that are cached according to [6].
3) $(N - N_1)$ least popular contents that are not cached at all.

In the delivery phase, each SBS receives Z_c number of requests. The cache of each SBS locally serves the content requests belonging to the first group, whereas the MBS server is responsible for the requests belonging to the second and third groups. In order to perform the coded caching scheme, the MBS has to maintain the requests for coded contents of each SBS separately. In this regard, As shown in Fig. 1, the MBS owns K distinct queues, where q_c^{coded}, $c \in \{1, 2, \dots, K\}$ denotes the queue that stores the requests of SBS c for coded contents. Moreover, requests are stored in q_c^{coded} by an arbitrary ordering. In addition, the MBS has one single queue which stores the requests of all SBSs for uncoded contents, denoted by q_{uncoded}.

We now explain the steps involved in the transmission of coded messages. Initially, MBS collects all head-of-line

(HoL) requests of the queues q_c^{coded} ($c \in \{1, 2, \ldots, K\}$) and then, in order to response to these requests, transmits the corresponding coded messages, following the scheme in [6]. The MBS then updates the HoL requests in the queues and repeats the same procedure, i.e., in step i, the ith rows of all queues are considered by the coded scheme. Note that the number of requests in q_c^{coded} could be less than Z_c since some of the requests of SBS c belong to either the first or the third group and thus are not stored in q_c^{coded}. Even requests belonging to the second group of contents might be repetitive and thus, are not stored in q_c^{coded} separately. Therefore, the number of queues involved in the coding process at step i could be less than K since some of the queues may not have any requests at step i. Moreover, the number of steps is at most $\max_{c\in\{1,\ldots,K\}}\{Z_c\}$, which happens when all requests of the SBS with the maximum number of users are distinct and associated with the coded contents. Finally, after sending all requested contents belonging to the second group with the coded scheme, the MBS sends contents related to all requests in q_{uncoded}, which guarantees that all users will be able to retrieve their requested contents.

Fig. 1 depicts a scenario where the number of SBSs is $K = 3$. The SBSs 1, 2, and 3 are responsible for 8, 4, and 6 users, respectively, and thus, they receive $Z_1 = 8$, $Z_2 = 4$, and $Z_3 = 6$ requests, respectively. The total number of contents is $N = 26$ and are ordered based on their popularity (i.e., 'A' is the most popular content). The cache size is $M = 5$ contents, and we assume that $M_1 = 3$, and $N_1 = 9$. As a result, the contents 'A', 'B', and 'C' are cached entirely. Moreover, 6 contents (from 'D' to 'I') are cached based on the coded caching scheme. The remaining 17 less popular contents (from 'J' to 'Z') are not cached. In the delivery phase, the uncoded data is highlighted in pink, while the contents in yellow are to be transmitted in the coded manner. The contents in green are locally hit by the local cache of SBSs, without any further transmission from the MBS. In this figure, the HoLs of all the queues during the first, second, and third steps in coded transmissions consist of $k_1 = 3$, $k_2 = 2$, and $k_3 = 1$ content requests, respectively.

4.2 Problem Formulation

The cost of traffic load is negligible in the placement phase; in contrast, the available bandwidth of the shared link becomes a bottleneck in the delivery phase and leads to a considerable cost. Hence, we ignore the traffic load in the placement phase and only consider the traffic load during the delivery phase. Let r be the expected traffic rate in each delivery phase, i.e., $r = E[R]$. Also, we denote by r_1 and r_2 as the expected traffic rates sent over the shared link at the delivery phase to satisfy requests from the second (i.e., coded) and third (i.e., uncoded) groups, respectively. Then we have $r = r_1 + r_2$. Our goal is to optimize the content placement in the SBSs' caches (i.e., to optimize M_1 and N_1) in order to minimize the expected traffic rate on the shared link during the delivery phase. Thus, the optimization problem is:

$$\min_{\substack{M\le N_1\le N\\ 0\le M_1\le M}} \{r\} \tag{2}$$

4.3 Performance analysis

In this sub-section, we determine the expected MBS traffic rate as a function of M_1 and N_1. We then characterize the optimum partitioning strategy as an optimization problem to find the minimum load. In the following, we assume that contents are sorted according to their popularities, i.e. $p_i \ge p_j$ if $i \le j$, (contents with a lower index are more popular). Also, the traffic rate contributed by the MBS is either related to the requests belonging to the second group of contents (i.e. coded contents with index from M_1+1 to N_1) or associated with requests belonging to the third group of contents (i.e. uncoded contents with index from N_1+1 to N). In this regard, in the following, we first derive the traffic related to each of these groups, separately, and then formulate the optimization problem.

Regarding the process explained before, we denote Q_i to be the random variable denoting the number of non-empty queues at step i, where $i = 1, 2, \ldots, \max(Z_1, ..., Z_K)$. In the following, the calculation of the MBS traffic rate is presented.

Lemma 1. *The traffic rate of the coded contents at step i given that $Q_i = k$, denoted by $r_1^{(i)}$, is derived as:*

$$r_1^{(i)} = \min\left(\frac{\binom{K}{T+1} - \binom{K-k}{T+1}}{\binom{K}{T}}, N_1 - M\right). \tag{3}$$

where $T = \frac{K\times(M-M_1)}{(N_1-M_1)}$.

Proof. See Appendix A for the proof. □

In the coded scheme proposed by [6], in the delivery phase, the coded messages are sent to all subsets of size $T+1$ of users which has requested a content. In Lemma 1, the term $\binom{K-k}{T+1}$ excludes those subsets that none of their members has requested a coded content. To calculate these subsets, in the following, we propose a lemma for deriving the probabilities that SBS c has more than i distinct coded requests.

Lemma 2. *Let $P_{c,i}$ denote the probability that the number of distinct coded requests of SBS c, denoted by l_c, is equal or greater than i, i.e., $P_{c,i} = Pr\{l_c \ge i\}$. $P_{c,i}$ is derived as follows:*

$$P_{c,i} = \sum_{j=i}^{Z_c} Pr\{l_c = j\}, \tag{4}$$

assume the $Pr\{l_c^{(z)} = j\}$ to be the probability of having j distinct coded requests in the first z requests in SBS c, where $z = 1, 2, \ldots, Z_c$, then:

$$Pr\{l_c = j\} = Pr\{l_c^{(Z_c)} = j\} \tag{5}$$

$Pr\{l_c^{(Z_c)} = j\}$ can be calculated with below recursive formula:

$$\begin{aligned}
&1) Pr\{l_c^{(0)} = 0\} = 1, Pr\{l_c^{(z)} = j | j > z\} = 0,\\
&2) Pr\{l_c^{(z)} = 0\} = Pr\{l_c^{(z-1)} = 0\} \times (1 - q_{c,1}),\\
&3) Pr\{l_c^{(z)} = j\} = Pr\{l_c^{(z-1)} = j\} \times (1 - q_{c,j+1})\\
&\quad + Pr\{l_c^{(z-1)} = j-1\} \times q_{c,j},
\end{aligned} \tag{6}$$

where $q_{c.j}$ is the probability that j-th distinct coded content is requested at SBS c, and is derived as follows (in this section we assume $p_{n,c} = p_n, \forall c \in \{1, \dots, K\}$):

$$q_{c,j} = q_j \begin{cases} = 0, & if \quad j > N_1 - M_1, \\ \simeq (1 - \frac{j-1}{N_1 - M_1}) \times \sum_{n=M_1+1}^{N_1} p_n, & otherwise. \end{cases} \tag{7}$$

Proof. See Appendix B for the proof. □

Using the above lemma, we derive the distribution of non-empty queues at each step of content delivery phase, where the HoL coded requests are responded.

Lemma 3. *Assume $Pr\{Q_i = k\}$ denotes the probability that exactly k SBSs request for coded contents at step i. It is derived as follows:*

$$Pr\{Q_i = k\} = Pr\{Q_i^{(K)} = k\} \tag{8}$$

where $Q_i^{(c)}$ is the random variable denoting the number of non-empty queues among the first c queues, i.e., $q_1^{coded}, \dots, q_c^{coded}$, at step i of coded caching. $Pr\{Q_i^{(K)} = k\}$ can be calculated with the following recursive equations:

$$\begin{aligned} &1) Pr\{Q_i^{(0)} = 0\} = 1, Pr\{Q_i^{(c)} = k | k > c\} = 0, \\ &2) Pr\{Q_i^{(c)} = 0\} = Pr\{Q_i^{(c-1)} = 0\} \times (1 - P_{c,i}), \\ &3) Pr\{Q_i^{(c)} = k\} = Pr\{Q_i^{(c-1)} = k\} \times (1 - P_{c,i}) \\ &\quad + Pr\{Q_i^{(c-1)} = k - 1\} \times P_{c,i}, \end{aligned} \tag{9}$$

where $P_{c,i}$ is derived from Lemma 2.

Proof. See Appendix C for the proof. □

In the following proposition, we derive the traffic rate of MBS.

Proposition 1. *Define $Z_{max} = \max_c Z_c$, then the expected traffic rate of coded content requests, denoted by r_1 is:*

$$r_1 = \begin{cases} \sum_{i=1}^{Z_{max}} \frac{\binom{K}{T+1} - \sum_{k=0}^{K} Pr\{Q_i = k\}\binom{K-k}{T+1}}{\binom{K}{T}}, & if \ N_1 > M, \\ 0, & otherwise \end{cases} \tag{10}$$

where $Pr\{Q_i = k\}$ is derived from lemma 3. Moreover, the expected traffic load of the uncoded content requests, denoted by r_2, is:

$$r_2 = \sum_{n=N_1+1}^{N} 1 - (1 - p_n)^{\sum_{c=1}^{K} Z_c}. \tag{11}$$

Finally, the total expected traffic rate is $r = r_1 + r_2$.

Proof. From Lemma 1, the expected traffic rate of the coded requests can be written as:

$$r_1 = E\left[\sum_{i=1}^{Z_{max}} \min\left(\frac{\binom{K}{T+1} - \binom{K-Q_i}{T+1}}{\binom{K}{T}}, N_1 - M\right)\right]. \tag{12}$$

Note that the expectation is taken over the random variables Q_i for $i = 1, 2, \dots, Z_{max}$. In (12), the minimization function involves two terms; the first term is maximized at $Q_i = K$ as follows:

$$\max\left(\frac{\binom{K}{T+1} - \binom{K-Q_i}{T+1}}{\binom{K}{T}}\right) = \frac{\binom{K}{T+1}}{\binom{K}{T}} = \frac{K-T}{T+1}. \tag{13}$$

By replacing T in the above equation, we have:

$$\frac{K-T}{T+1} = \frac{K \times (N_1 - M)}{N_1 - M_1 + K \times (M - M_1)}. \tag{14}$$

Letting (14) be less than $N_1 - M$, i.e., the second argument in the min function in (12), leads to:

$$1 < \frac{N_1 - M_1}{K} + (M - M_1),$$

which always holds in the case of $N_1 > M$ and $M - M_1 \geq 1$. Thus, the first argument of min in (12) is selected. Also, in the case that $N_1 = M$, and therefore $M_1 = M$ (pure uncoded caching), the second argument of min in (12) is selected, and thus we have $r_1 = 0$. Therefore, when $N_1 > M$ (and so $M_1 < M$), (12) reduces to:

$$\begin{aligned} r_1 &= E\left[\sum_{i=1}^{Z_{max}} \frac{\binom{K}{T+1} - \binom{K-Q_i}{T+1}}{\binom{K}{T}}\right] \\ &= \sum_{i=1}^{Z_{max}} \frac{\binom{K}{T+1} - E\left[\binom{K-Q_i}{T+1}\right]}{\binom{K}{T}} \\ &= \sum_{i=1}^{Z_{max}} \frac{\binom{K}{T+1} - \sum_{k=0}^{K} Pr\{Q_i = k\}\binom{K-k}{T+1}}{\binom{K}{T}}. \end{aligned} \tag{15}$$

Next we prove (11). As mentioned before, any request (in all K SBSs) from the third group of contents that is not cached (contents indexed from $N_1 + 1$ to N) should be satisfied directly by the MBS. However, if the MBS receives multiple requests for specific content in a time slot, it uses broadcasting to send the content only once. Hence, the expected traffic load of such uncached contents is equal to the expected number of distinct requests for them. The probability that content n is not requested is $(1 - p_n)^{\sum_{c=1}^{K} Z_c}$. Therefore, the probability that content n is requested at least one time is $1 - (1 - p_n)^{\sum_{c=1}^{K} Z_c}$. Thus, the expected total number of distinct requests of uncoded contents is equal to the sum of aforementioned expected probability for all uncoded contents, as indicated in (11). This completes the proof. □

We now formulate the optimum partitioning problem in order to minimize the traffic rate from MBS to the SBSs, i.e., r. The minimization problem is formulated as follows:

$$\begin{aligned} &\min_{\substack{M \leq N_1 \leq N \\ 0 \leq M_1 \leq M}} \{r_1 + r_2\} \\ &s.t. \\ &T = \frac{K \times (M - M_1)}{(N_1 - M_1)} \in \mathbb{N}. \end{aligned} \tag{16}$$

If $N_1 > M$ and $M_1 = 0$, then (16) is reduced to the optimization of the two-partitioning pure coded scheme. Also, it can be proved that when the popularity of contents is the same for all SBSs, then caching the M most popular contents in SBSs is the optimal placement strategy of the pure uncoded scheme. In other words, if $N_1 = M$, then (16) is reduced to the optimal pure uncoded scheme.

The optimal value of (16) can be found in polynomial time by exhaustive search. However, in order to expedite the search process, we consider the following points:

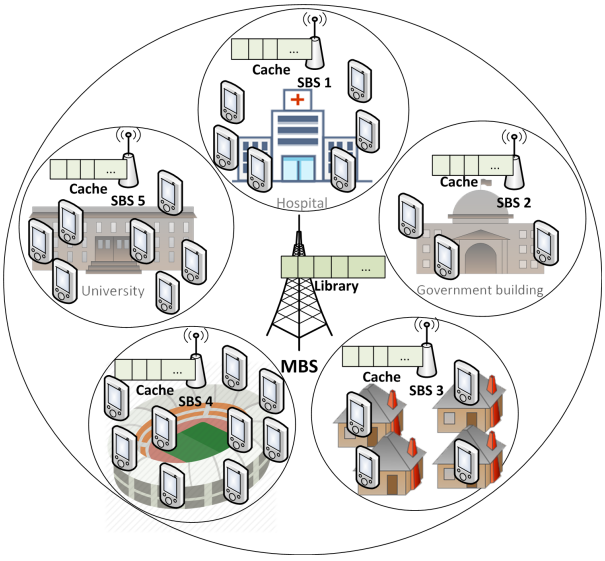


Fig. 2: An example of SBS-dependent (heterogeneous) content popularity.

- We only consider those values of N_1 and M_1 which result in $T = \frac{K\times(M-M1)}{(N1-M1)}$ to be an integer value.
- By considering $\frac{\binom{K}{T+1}}{\binom{K}{T}} = \frac{K-T}{T+1}$, we calculate $\frac{K-T}{T+1}$ only once for each possible N_1 and M_1 values.
- As mentioned before, by defining an array and storing the result of earlier computations (dynamic programming), we can compute (9) and (6) in polynomial time. We can also use this technique to skip duplicate computations in other parts of the problem; for example, storing the result of (11) for one value of N_1 could be used to compute it for the next N_1 values with less computing.

5 SBS-DEPENDENT NON-UNIFORM CONTENT POPULARITY

In the previous section, we assumed that the content popularity distribution and the size of the caches are the same for all SBSs. However, in this section, we consider a more general case and assume heterogeneous content popularity distribution and different cache sizes. It is worth noting that according to the reports, the global popularities are not a good representation of the local preferences of the users. For example, a two-week study in 1,000 locations of a metropolis in China, where nearly 2 million users use mobile devices to view more than 300 thousand unique videos, indicates that videos have quite different popularities in different locations. As such, it is reported that in 60% of locations, the average popularity of the 1000 most globally popular videos (top 0.3% of all viewed videos) is below the average popularity of the top 40% locally popular videos [37]. Therefore, we assume that users in the range of one SBS share similar preferences, and there may be several popularity groups inside the coverage area of an MBS. While some contents may be popular for almost all groups of users, some others may be popular only for specific groups.

As an example, Fig. 2, illustrates an MBS with five SBSs under its coverage, where the SBSs serve groups with different preferences. If the coventional uncoded caching is used in such a scenario, i.e., the SBSs cache the contents which are globally popular in their caches, then MBS has to send the local popular contents of SBSs repeatedly, leading to high traffic loads. However, if the SBSs store their local popular contents, then MBS only transmits the least popular contents, taking advantage of broadcasting. Also in the case of applying coded caching, choosing contents which are only popular in one SBS and not other SBSs leads to the waste of cache memory in other SBSs, and consequently increases the traffic load. Therefore, it is crucial to consider the heterogeneous SBS-dependent popularities for designing the caching strategy. On the other hand, SBSs may share similar preferences, e.g., in Fig. 2, the users of SBS 4 are interested in sports, politic events are more popular in SBSs 2 and 5, and users of SBS 3 follow both politics and sports. Based on these similarities, we can form two groups of SBS, a group composed of SBSs 3 and 4 and another one composed of SBSs 2,3, and 5.

In this section, we extend the hybrid caching scheme of the previous section to the case of SBS-dependent popularities and non-equal cache sizes. In particular, unlike the previous section, where the parameters N_1 and M_1 are the same for all SBSs, in this section, they are optimized considering possibility of different groups.

5.1 The proposed SBS-dependent caching scheme

In this part, we define the caching strategy in the case of heterogeneous popularities and non-equal cache sizes, i.e., when $p_{n,c}$ and M_c is not the same for every SBS c, as follows. The cache of each SBS c is divided into two part; uncoded and coded. The capacity of uncoded part in SBS c is denoted by M_{1c}, i.e., M_{1c} contents are cached entirely in the uncoded part of SBS c. We let $Y_{n,c} = 1$ if content n is cached uncoded at SBS c and $Y_{n,c} = 0$ otherwise. Consequently, we have $\sum_n Y_{n,c} = M_{1c}$, and $M_c - M_{1c}$ of cache capacity is left for coded part. On the other hand, we define groups of SBSs, where the SBSs within a group share similar preferences and thus, participate in the same coded caching scheme. It is worth noting that a single SBS may participate in multiple groups since while a part of its preferences are common in a group, other parts may be popular in other groups. In this regard, the grouping scheme of SBSs are defined as follows. We define $G = \{G_1, G_2, ..., G_{|G|}\}$ to be a cover of S (the set of SBSs), where $G_i \subset S$ with $|G_i| \geq 2$ and $\cup_{i=1}^{|G|} G_i = S$. According to this definition, we have $|G|$ groups of SBSs in cover G which they have at least two members (for the sake of applying coded caching scheme), may be overlapping, and cover all SBSs. Moreover, we let $S_{c,g} = 1$ if SBS c participate in group $g \in G$, and $S_{c,g} = 0$ otherwise. Then, the number of SBSs in group g, denoted by K_g, is derived as $\sum_c S_{c,g} = K_g$.

A separate coded caching scheme is applied in each group. Since SBS c may participate in multiple groups, it dedicates a part of its cache to each of its participating group. In particular, the capacity M_g is dedicated to group g, if $S_{c,g} = 1$. Moreover, we let $X_{n,g} = 1$ if content n participates in the coded scheme of group g and $X_{n,g} = 0$, otherwise. Consequently, the number of contents participat-

ing in coded scheme of group g, denoted by N_g, is derived as $\sum_n X_{n,g} = N_g, \forall g$.

It is worth noting that in the proposed caching strategy for the heterogeneous case, unlike the homogeneous case, each SBS caches M_{1c} uncoded contents that are different from other SBSs. Also, while the dedicated cache capacity for coded contents remains the same inside a group, but in general in each SBS, a different capacity is dedicated to coded contents. As such, the dedications in each SBS should satisfy $M_{1c} + \sum_g S_{c,g} \times M_g = M_c$. Moreover, since $|G|$ concurrent coded schemes are applied, MBS should maintain the different set of coded queues for each of these groups. In this regard, the queues $q_{1,g}^{coded}, \ldots, q_{K_g,g}^{coded}$ are the queues dedicated to group g. Finally, a specific content W_n may be cached uncoded in SBS c, while it also be included in coded scheme of some of the participating groups. Similar to Section 4, In this section, we aim to optimize $r_1 + r_2$ for the proposed scenario.

5.2 Performance analysis

In the following, we derive the traffic rate of the MBS under the proposed caching strategy for SBS-dependent non-uniform content popularity distribution.

Lemma 4. *Let $P_{c,g,i}$ denote the probability that the number of distinct coded requests of SBS c in the cluster g, denoted by $l_{c,g}$, is equal or greater than i, i.e., $P_{c,g,i} = Pr\{l_{c,g} \geq i\}$. $P_{c,i,g}$ is derived as follows:*

$$P_{c,g,i} = \sum_{j=i}^{Z_c} Pr\{l_{c,g} = j\}. \tag{17}$$

Also, let $Pr\{l_{c,g}^{(z)} = j\}$ be the probability of having j distinct coded requests in first z requests in SBS c, where $z = 1, 2, \ldots, Z_c$, then:

$$Pr\{l_{c,g} = j\} = Pr\{l_{c,g}^{(Z_c)} = j\}, \tag{18}$$

where $Pr\{l_{c,g}^{(Z_c)} = j\}$ can be calculated with below recursive formula:

$$\begin{aligned}
&1) Pr\{l_{c,g}^{(0)} = 0\} = 1, Pr\{l_{c,g}^{(z)} = j | j > z\} = 0,\\
&2) Pr\{l_{c,g}^{(z)} = 0\} = Pr\{l_{c,g}^{(z-1)} = 0\} \times (1 - q_{c,g,1}),\\
&3) Pr\{l_{c,g}^{(z)} = j\} = Pr\{l_{c,g}^{(z-1)} = j\} \times (1 - q_{c,g,j+1})\\
&\quad + Pr\{l_{c,g}^{(z-1)} = j-1\} \times q_{c,g,j},
\end{aligned} \tag{19}$$

where $q_{c,g,j}$ is approximated to be:

$$q_{c,g,j} \begin{cases} = 0, & if \quad j > N_g, \\ \simeq (1 - \dfrac{j-1}{N_g}) \times \sum_{n=1}^{N} X_{n,g}.p_{n,c} & otherwise. \end{cases} \tag{20}$$

Proof. See Appendix D for the proof. □

Lemma 5. *Let $Pr\{Q_{i,g} = k\}$ be the probability that exactly k SBSs in cluster g request for coded contents at step i . Then, we have:*

$$Pr\{Q_{i,g} = k\} = Pr\{Q_{i,g}^{(K_g)} = k\}, \tag{21}$$

where $Q_{i,g}^{(c_g)}$ is the random variable denoting the number of non-empty queues among the first c_g queues in cluster g, i.e., $q_{1,g}^{coded}, \ldots, q_{c_g,g}^{coded}$, at step i of coded caching. $Pr\{Q_{i,g}^{(K_g)} = k\}$ can be calculated with the following recursive equations:

$$\begin{aligned}
&1) Pr\{Q_{i,g}^{(0)} = 0\} = 1, Pr\{Q_{i,g}^{(c_g)} = k | k > c_g\} = 0,\\
&2) Pr\{Q_{i,g}^{(c_g)} = 0\} = Pr\{Q_{i,g}^{(c_g-1)} = 0\} \times (1 - P_{c_g,g,i}),\\
&3) Pr\{Q_{i,g}^{(c_g)} = k\} = Pr\{Q_{i,g}^{(c_g-1)} = k\} \times (1 - P_{c_g,g,i})\\
&\quad + Pr\{Q_{i,g}^{(c_g-1)} = k-1\} \times P_{c_g,g,i},
\end{aligned} \tag{22}$$

where $P_{c_g,g,i}$ is derived from lemma 4.

Proof. The proof of this lemma is similar to the proof of lemma 3, except that in this lemma, there are G coded delivery groups. Therefore, the equations are calculated for each group separately. However, for each group, only the SBSs that participate in it are considered. □

Proposition 2. *If $T_g = \frac{K_g \times M_g}{N_g}$, and $Z_{max} = \max_c Z_c$, then the expected traffic rate of coded content requests, denoted by r_1, is:*

$$r_1 = \sum_{g \in G} \times \begin{cases} \sum_{i=1}^{Z_{max}} \frac{\binom{K_g}{T_g+1} - \sum_{k=0}^{K_g} Pr\{Q_{i,g}=k\} \binom{K_g-k}{T_g+1}}{\binom{K_g}{T_g}}, & if\ N_g > M_g, \\ 0, & otherwise \end{cases} \tag{23}$$

where $Pr\{Q_{i,g} = k\}$ is derived from lemma 5. Moreover, the expected traffic rate of uncoded content requests, denoted by r_2, is:

$$r_2 = \sum_{n=1}^{N} \left(1 - \prod_{c=1}^{K} \left(1 - (p_{n,c} \times (1 - Y_{n,c}) \times \prod_{g \in G} (1 - X_{n,g}.S_{c,g}))\right)^{Z_c}\right). \tag{24}$$

Finally, the total expected traffic rate is $r = r_1 + r_2$.

Proof. Equation (23) of this proposition is similar to equation (10) of proposition 1, except that in this proposition, there are G coded delivery groups. Therefore, the coded rate of the MBS is calculated for each group separately. Finally, the total coded rate of the MBS is the summation of calculated coded rates of all groups. In order to prove (24), we should consider that if content W_n is cached neither uncoded at SBS c, i.e., $Y_{n,c} = 0$, nor coded, i.e., $X_{n,g} \times S_{c,g} = 0, \forall g \in G$, then content W_n will be responded by MBS if it is requested in SBS c. Consequently, the probability that content W_n is not requested by SBS c from MBS is equal to: $\left(1 - (p_{n,c} \times (1 - Y_{n,c}) \times \prod_{g \in G} (1 - X_{n,g}.S_{c,g}))\right)^{Z_c}$. If at least one SBS requests W_n, this content will be broadcast by MBS once and thus contributes to the traffic rate r_2. This completes the proof. □

Let $\widetilde{G}$ denote the set of all possible covers of SBSs ($|\widetilde{G}| = 2^K - K - 1$).Then the optimum partitioning problem with

the objective of minimizing the traffic rate from the MBS to SBSs is written as follows:

$$\min_{G\subset\bar{G}}\{\min_{\substack{0\le M_{1c}\le M_c\\ 1\le M_g\le M\\ M_g<N_g\le N\\ X_{n,g},Y_{n,c}\in\{0,1\}}}\{r_1+r_2\}\}$$

$$s.t.$$

$$T_g=\frac{K_g\times M_g}{N_g}\in\mathbb{N},$$

$$\sum_{n=1}^{N}X_{n,g}=N_g,\forall g\in G.$$

$$\sum_{n=1}^{N}Y_{n,c}=M_{1c},\forall c\in\{1,\dots,K\}.$$

$$M_{1c}+\sum_{g\in G}M_g\times S_{c,g}=M_c,\forall c\in\{1,\dots,K\}. \quad (25)$$

As can be seen in (25), finding the optimal placement strategy for the hybrid scheme is intractable. Even if the optimal covering (G), the memory allocations (M_g) and the number of contents involved in each group (N_g) are specified then in order to find the best content placement, we need to calculate the MBS rate r for $\prod_{c=1}^{K}\binom{N}{M_{1c}}\times\prod_{g=1}^{G}\binom{N}{N_g}$ possible configuration. A special case of this problem is when there is no significant similarity in the content popularity among the SBSs. In this case, increasing the number of coded delivery groups only reduces the global cache gain, and therefore the optimal cover ($G^\star$) will have one member which is the set of all SBSs. Here, the optimal configuration is similar to the previous section, in which the SBSs' caches are divided into two parts. However, the difference is that the uncoded contents of one SBS can be different from other SBSs. Also, although the coded contents for all SBSs are the same, they are not necessarily the same as those would be selected based on global popularity. Besides, to find the best content placement for this special case, we need to calculate the MBS rate r for $\binom{N}{N_1}\times\prod_{c=1}^{K}\binom{N}{M_{1c}}$ possible configuration. Even in the cases of the two-partitioning pure coded, [26]–[30] and pure uncoded schemes we need to check $\binom{N}{N_1}$ and $\prod_{c=1}^{K}\binom{N}{M_c}$ possible configurations respectively to find the optimal content placement.

6 Numerical Results

In this section, the performance of the proposed hybrid scheme is evaluated and compared with the baseline pure coded and conventional uncoded schemes and previous works, as well as the two-partitioning scheme reported in [26]–[30], through numerical evaluations and simulation. We validate analytical results with the simulations conducted using MATLAB for a period of 2000 time slots. In the following, we first evaluate the proposed hybrid scheme for SBS-independent non-uniform popularity distribution under a heterogeneous number of demands, and then we consider the SBS-dependent popularity distribution.

6.1 SBS-independent non-uniform popularity distribution

In this subsection, for analytical results, the optimum placement strategy of the hybrid scheme for integer values of $T=K\times(M-M_1)/(N_1-M_1)$ is obtained from (16). Also, we suppose that the content popularity distribution follows the Zipf popularity profile with parameter $\alpha>0$ as follows:

$$p_n=\frac{(\frac{1}{n})^\alpha}{\sum_{j=1}^{N}(\frac{1}{j})^\alpha}. \quad (26)$$

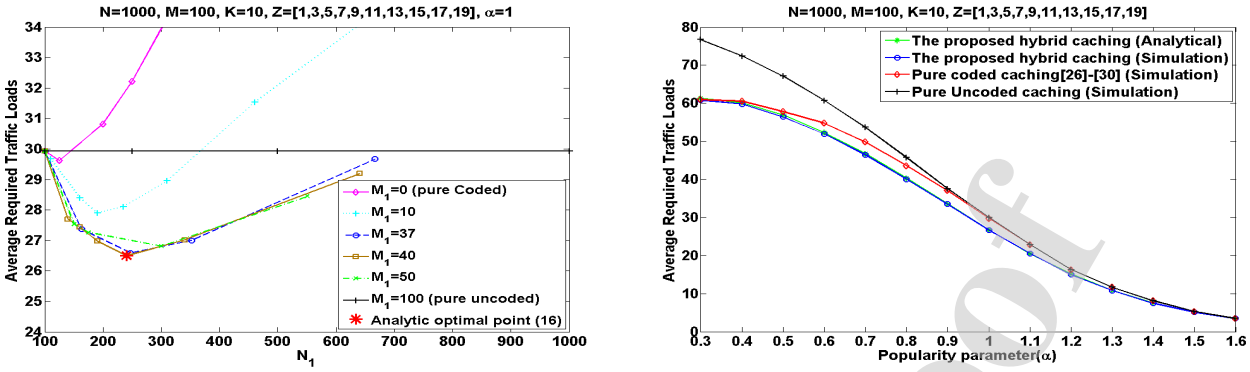
In the following, we first verify the approximation used for finding the optimal configuration of the hybrid scheme. We then study the effect of content popularity, the standard deviation of the number of users in SBSs, and the system scale on the optimum traffic load of the shared link.

Fig. 3a shows the simulation results of MBS traffic load versus N_1 for different scenarios of the proposed cache partitioning as well as pure coded [26]–[30] and pure uncoded caching schemes. The simulation results of different cache partitionings (i.e., different values of M_1 and N_1) show that the approximate analytical optimal partitioning found by the optimization problem in (16) is very close to the optimal partitioning derived from simulations. These results indicate that the approximation that we used to find the optimal partitioning is suitable. As can be understood from (16) and Fig. 3a, the hit ratio of local cache improves as M_1 increases. But on the contrary, because of the reduction of memory space for the coded section, the bandwidth load required to satisfy the coded content requests increases. Moreover, by increasing the value of N_1, although the contents that are not cached decrease, the required bandwidth load to satisfy requests of coded contents rises.

In Figs. 3b and 4, for the hybrid and pure coded schemes [26]–[30], we first find the M_1^* and N_1^* values for each parameter settings that minimize the MBS traffic load. The caching schemes are then evaluated for their corresponding optimal configuration via simulation. Fig. 3b illustrates the simulation and analytical results for the traffic load as a function of the practical Zipf parameter in the interval $\alpha\in[0.5,1.6]$ [39]. TABLE 2 shows the optimal configuration of the hybrid caching for some different scenarios of user distribution in the SBSs where there are 100 users in the system ($K=10$). Fig. 4a also depicts the traffic load of the MBS as a function of the standard deviation of the number of users in the SBSs. This figure shows a comparison of the optimal placement results of different schemes for the configurations of TABLE 2. It is evident from these figures that the simulation results are very close to the analytical

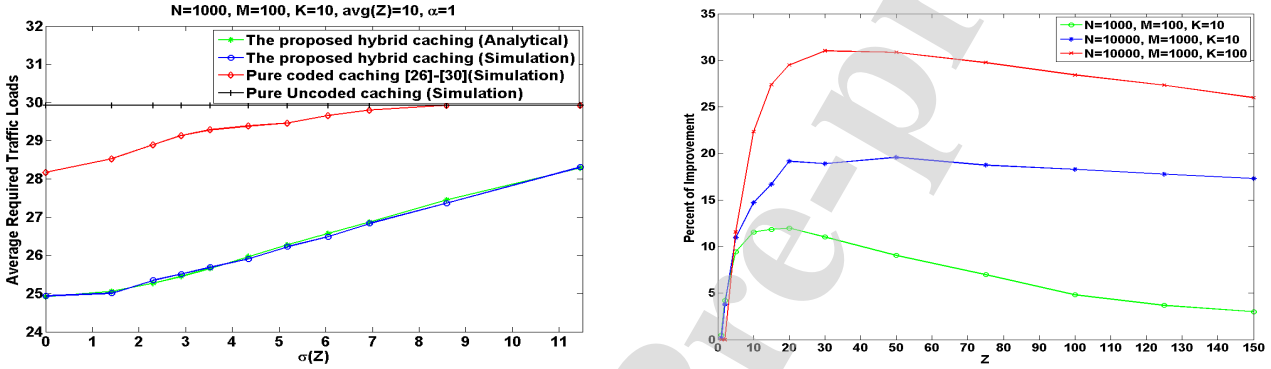
TABLE 2: Optimal configuration (N_1^* and M_1^*) of the hybrid scheme for different users distribution in SBSs, where $K=10, N=1000, M=100, \alpha=1$ and there are 100 users in the system.

No. of users (Z_1, ... ,Z_{10})	σ(Z)	N_1^*, M_1^*
10 10 10 10 10 10 10 10 10 10	0.0	352, 37
8 9 9 9 9 10 11 11 12 12	1.4142	344, 39
6 8 9 9 9 10 11 12 12 14	2.3094	340, 40
5 7 9 9 9 10 11 12 13 15	2.9059	332, 42
4 6 9 9 9 10 11 12 14 16	3.5277	328, 43
3 5 7 9 9 11 11 13 15 17	4.3461	316, 46
2 4 6 8 9 11 12 14 16 18	5.1854	240, 40
1 3 5 7 9 11 13 15 17 19	6.0553	240, 40
0 2 4 6 9 11 14 16 18 20	6.9442	233, 43
0 2 2 3 7 11 14 16 20 25	8.5894	219, 49
1 1 1 1 1 5 15 20 25 30	11.4504	172, 52



(a) (b)

Fig. 3: MBS Traffic load as a function of (a) N_1 for different M_1 (b) popularity parameter.



(a) (b)

Fig. 4: (a) MBS traffic load as a function of standard deviation of Z (b) percent of improvement (offloading MBS traffic load) of the proposed hybrid caching compared to the two-partitioning pure coded methods [26]–[30].

findings, and the hybrid caching can lead to significant traffic off-loading compared to the two-partitioning pure coded [26]–[30] and pure uncoded schemes.

As can be seen in Fig. 4a, when the standard deviation of the number of users in the SBSs increases, the performance of the coded schemes is reduced. This is because the number of coded requests in the corresponding queues of the MBS becomes very unbalanced. As a result, the expectation of the number of sending steps for a specific configuration increases, and the optimal configuration, as shown in TABLE 2, tends to a smaller quantity of N_1 and larger quantities of M_1. Fig. 4b shows the percent of improvement of the hybrid scheme compared to the two-partitioning pure coded methods [26]–[30] in terms of MBS traffic load versus the number of users within the coverage of each SBS for three different system scales (content library sizes, cache capacities, and number of SBSs). In this figure, the number of users of all SBSs is assumed to be the same and equal to Z. As can be seen, the hybrid scheme has made significant improvement in the MBS traffic load, especially for $Z > 2$. In addition, when Z increases, this percent of improvement increases at first but then decreases. This is because in the hybrid scheme, by increasing Z, some requests are hit in the M_1 part of SBSs' cache, and therefore the number of steps of sending coded messages becomes significantly less than the pure coded methods. But when Z increases further, the probability of duplicate requests also increases (especially for smaller library sizes). Therefore, the number of steps of sending coded messages in pure coded methods, and therefore the percentage of improvement is reduced.

6.2 SBS-dependent non-uniform popularity distribution

In this subsection, we suppose the content popularity distribution is not the same for different SBSs. In Fig. 5, we suppose that the number of available contents in the MBS is $N = 4$, there are a total of $K = 4$ SBSs, each SBS only serves one user ($Z_c = 1\ \forall c \in \{1, \dots, K\}$), and the content popularity distribution is according to the TABLE 3. The results are depicted for various cache capacity of SBSs ($M = 1, 2$ and 3). As can be seen in this figure, the hybrid scheme offloads more traffic compared to the two-partitioning pure coded and pure uncoded schemes.

In particular, when $M = 2$, the hybrid scheme is better than both other schemes. In this condition, the best configuration of the hybrid scheme is $N_1^* = 3$ and $M_1^* = 1$, where (W_3) is stored entirely in the caches of $SBS1\&2$, and W_4 is stored entirely in the caches of the $SBS3\&4$, also (W_1, W_2) are stored with the coded scheme in the caches of all SBSs. Whereas optimal placement of the two-partitioning pure coded method is $N_1^* = 4$, or in other words, all $W_1 \dots W_4$ are cached with the coded scheme in the caches of all SBSs. Besides, the optimal placement of pure uncoded scheme is

TABLE 3: Content Popularity distribution for $N = 4$ and $K = 4$ scenario.

	W_1	W_2	W_3,	W_4
$SBS1$	0.3	0.2	0.5	0.0
$SBS2$	0.2	0.3	0.5	0.0
$SBS3$	0.3	0.2	0.0	0.5
$SBS4$	0.2	0.3	0.0	0.5

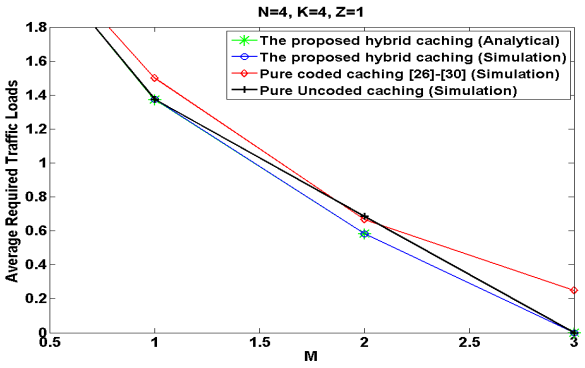


Fig. 5: MBS traffic load as a function of M.

caching W_1 (or W_2) in all SBSs, W_3 in $SBS1\&2$, and W_4 in $SBS3\&4$. In the pure uncoded scheme, one of W_1 or W_2 is cached in all SBSs, and for the other one, it benefits from multicast opportunities.

7 Conclusion and Future Works

In this paper, we have studied content caching for the shared medium networks and have proposed a hybrid coded-uncoded caching under heterogeneous users' behaviors. In practice, the proposed scenario corresponds to a cellular network that includes an MBS and multiple SBSs where each SBS is equipped with a limited size cache and serves multiple users. In particular, we assume that each SBS can request a different number of contents than the other ones. Also, we consider non-uniform content popularity distribution, which can be the same for all SBSs (SBS-independent) or different for each one (SBS-dependent). We derive explicit closed-form expressions for the server load of the proposed hybrid caching at the delivery phase and formulate the optimum cache partitioning problem. Validated by simulation results, our findings showed that the proposed scheme outperforms the baseline schemes of pure uncoded and pure coded caching, as well as the two-partitioning scheme existing in the literature. For future topics, we first plan to propose heuristic and machine-learning methods in the case of SBS-dependent non-uniform popularity distribution to find appropriate configuration of the proposed hybrid scheme in polynomial complexity. Then, we try to extend this work for online caching problems.

Appendix A
Lemma1 Proof

Proof. Assume that each SBS has one request and all contents are to be cached coded. Then, according to the coded caching scheme in [6], each content is split into $\binom{K}{T}$ non-overlapping fragments, each of size $f/\binom{K}{T}$, where $T = \frac{KM}{N}$. Also, each cache selects a T/K fraction of all fragments. Then, in each transmission, MBS chooses a new subset of SBSs with size $T+1$, chooses one fragment from the content requested by each SBS, and then, XORs $T+1$ chosen fragments and transmits that. This procedure continues until all possible subsets are chosen. Note that in each transmission, the fragment which is missing at one SBS is available at all other T SBSs (see the cache placement phase in [6]). Consequently, in each transmission, each selected SBS is able to decode one missing fragment using the received signal as well as its cached content. Finally, after $\binom{K}{T+1}$ multicast transmissions, each SBS retrieves its requested content completely. But as mentioned earlier, in step i of coded transmissions in our problem, all SBSs do not have necessarily request, i.e., the coded queues of some SBSs may be empty. In this case, no coded messages are transmitted to those subsets of SBSs that none of their members has a request. Hence, if the number of caches that have requests for the coded contents in step i equals k, then the number of unnecessary transmissions is $\binom{K-k}{T+1}$. Consequently, the number of multicast transmissions at this step is $\binom{K}{T+1} - \binom{K-k}{T+1}$, where the size of each transmission is equal to $F/\binom{K}{T}$ and $T = \frac{K\times(M-M_1)}{(N_1-M_1)}$. The latter is due to the fact that the cache and library sizes are $M - M_1$ and $N_1 - M_1$, respectively.

Contrarily, if $(N_1 - M_1) < k$, then the MBS enjoys an improvement of $(N_1 - M_1)/k$ from broadcast. Thus, from (1), the traffic load of coded contents is given as:

$$k \times \left(1 - \frac{M - M_1}{N_1 - M_1}\right) \times \frac{N_1 - M_1}{k} = N_1 - M. \qquad (27)$$

This completes the proof. □

Appendix B
Lemma2Proof

Proof. If l_c denotes the number of distinct coded requests of SBS c, then, $P_i^{(c)}$ equals to $Pr\{l_c >= i\}$ and can be calculated according to (4). Due to the possibility of duplicate coded requests, the native approach to calculate $Pr\{l_c = j\}$ has exponential complexity. However, an approximation of it can be calculated with recursion and dynamic programming in polynomial complexity as fallowing: if the $Pr\{l_c^{(z)} = j\}$ denotes the probability of having j distinct coded requests in first z requests in SBS c, where $z = 1, 2, \ldots, Z_c$, then $Pr\{l_c = j\}$ is equal to $Pr\{l_c^{(Z_c)} = j\}$ and can be calculated with a recursive formula that is given in (6), where the notation q_j^c denotes the probability of requesting the jth distinct coded content at the next request in SBS c, where $j-1$ distinct coded contents have been requested until this request. Therefore, q_j^c only depends on the popularity of coded contents and previous $j-1$ distinct requested coded contents. On the other hand, calculating q_j^c is independent of all number of requests. In this section, we assume that the content popularity distribution is SBS-independent in other words, $p_{n,c} = p_n$, $\forall c \in \{1, \ldots, K\}$. Therefore, based on the above definition, q_j^c is the same for all SBSs and hence $q_j^c = q_j, \forall c \in \{1, \ldots, K\}$.

Due to the non-uniform popularity distribution of the contents and possibility of duplicate requesting the previous $j-1$ requested coded contents, calculating the exact amount

of q_j is highly complicated. To this aim, we approximate q_j by approximating the probability of requesting $j-1$ previous contents as shown in (7). Apparently, when the popularity distribution is almost uniform, this approximation is more accurate. Contrarily, when the popularity distribution is extremely non-uniform, where few contents are in high demand, this approximation yields lower accuracy and may be calculating a bit larger value for the r_1. However, under extremely non-uniform popularity distributions, the hybrid caching strategy tends to cache these high demand contents entirely (tends to a higher value for M_1), and this approximation may reinforce this tendency. By caching the most popular contents entirely, the error of this approximation is greatly reduced. Besides, our goal is to find the best library partitioning policy rather than calculating the exact rate. Therefore, the possible error of such approximation does not affect rate calculation for all contents but only for contents located in different partitions based on different policies. Hence, as the negligible impact of this approximation on our choice of library partitioning, i.e., N_1 and M_1, will also be shown numerically and via simulation in the following sections, this approximation is reasonable and has minimal impact on choosing the optimal policy. However, using this approximation is not recommended for selecting the best caching policy for pure coded methods.

This completes the proof. □

APPENDIX C
LEMMA3PROOF

Proof. $\{Q_i = k\}$ occurs when exactly k SBSs receive at least i distinct requests for the coded contents, while the rest of the SBSs receive less than i distinct requests for these contents. Since the number of requests of each SBS may be different from other SBSs, the probability of requesting the coded contents is different for different SBS. Therefore, calculating the $Pr\{Q_i = k\}$ with the native approach is a combinatorial problem with exponential complexity. However, we can calculate $Pr\{Q_i = k\}$ in polynomial complexity by using recursion and dynamic programming as follows. If the $Pr\{Q_i^{(c)} = k\}$ denotes the probability of having k caches with coded requests in first c caches, where $c = 1, 2, \ldots, K$, then as it is shown in (8), the $Pr\{Q_i = k\}$ is equal to $Pr\{Q_i^{(K)} = k\}$, and it can be calculated with the recursive formula given in (9), where the notation $P_i^{(c)}$ denotes the probability that SBS c (which receives Z_c request at each time slot from Z_c users) has at least i distinct requests for coded contents.

This completes the proof. □

APPENDIX D
LEMMA4PROOF

Proof. The proof of lemma4 is similar to lemma2, and the proof of equations (17)-(19) of this lemma is similar to lemma2, except that in lemma4, there are G coded delivery clusters, and therefore these equations are calculated for each one separately. However, for each cluster, only the SBSs that participate in it are considered. In addition, in lemma4, the $p_{n,c}$ is not the same for all SBSs. Therefore equation (20) has some differences from (7). However, likewise lemma2, $q^c_{g,j}$ for each cluster g only depends on the popularity of N_g contents and previous $j-1$ requests for these contents from corresponding SBS c. Also, $q^c_{g,j}$ is calculated by approximating the probability of requesting the previous $j-1$ of N_g contents in SBS c. Except that in this lemma, $q^c_{g,j}$ is not the same for different SBSs, and also for each SBS, it is not the same for different clusters. Note that for each cluster g only K_g SBSs that participate in it are considered. Therefore, $q^c_{g,j}$ is calculated only for SBSs, which are participated in cluster g. As shown in (20), The array X is used to determine coded contents corresponding to cluster g.

This completes the proof. □

REFERENCES

[1] A. G. Sheshjavani, A. Khonsari, S. P. Shariatpanahi, M. Moradian, and A. Dadlani, "Coded caching under non-uniform content popularity distributions with multiple requests," in *2020 IEEE Wireless Communications and Networking Conference (WCNC)*, 2020, pp. 1–6.

[2] V. Chandrasekhar, J. G. Andrews, and A. Gatherer, "Femtocell networks: a survey," *IEEE Communications Magazine*, vol. 46, no. 9, pp. 59–67, Sep. 2008.

[3] K. Shanmugam, N. Golrezaei, A. G. Dimakis, A. F. Molisch, and G. Caire, "Femtocaching: Wireless content delivery through distributed caching helpers," *IEEE Transactions on Information Theory*, vol. 59, no. 12, pp. 8402–8413, Dec 2013.

[4] 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.

[5] Y. Chen, M. Ding, J. Li, Z. Lin, G. Mao, and L. Hanzo, "Probabilistic small-cell caching: Performance analysis and optimization," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 5, pp. 4341–4354, May 2017.

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

[7] N. Karamchandani, U. Niesen, M. A. Maddah-Ali, and S. Diggavi, "Hierarchical coded caching," in *2014 IEEE International Symposium on Information Theory*, June 2014, pp. 2142–2146.

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

[9] M. A. Maddah-Ali and U. Niesen, "Decentralized coded caching attains order-optimal memory-rate tradeoff," *IEEE/ACM Transactions on Networking*, vol. 23, no. 4, pp. 1029–1040, Aug 2015.

[10] M. K. Kiskani and H. R. Sadjadpour, "Throughput analysis of decentralized coded content caching in cellular networks," *IEEE Transactions on Wireless Communications*, vol. 16, no. 1, pp. 663–672, Jan 2017.

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

[12] R. Pedarsani, M. A. Maddah-Ali, and U. Niesen, "Online coded caching," *IEEE/ACM Transactions on Networking*, vol. 24, no. 2, pp. 836–845, April 2016.

[13] 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.

[14] S. B. Hassanpour, A. Khonsari, S. P. Shariatpanahi, and A. Dadlani, "Hybrid coded caching in cellular networks with d2d-enabled mobile users," in *2019 IEEE 30th Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, Sep. 2019, pp. 1–6.

[15] Y. Lu, W. Chen, and H. V. Poor, "Coded joint pushing and caching with asynchronous user requests," *IEEE Journal on Selected Areas in Communications*, vol. 36, no. 8, pp. 1843–1856, Aug 2018.

[16] Y. Wei and S. Ulukus, "Coded caching with multiple file requests," in *2017 55th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, Oct 2017, pp. 437–442.

[17] M. Ji, A. Tulino, J. Llorca, and G. Caire, "Caching-aided coded multicasting with multiple random requests," in *2015 IEEE Information Theory Workshop (ITW)*, April 2015, pp. 1–5.

[18] H. Xu, C. Gong, and X. Wang, "Efficient file delivery for coded prefetching in shared cache networks with multiple requests per user," *IEEE Transactions on Communications*, vol. 67, no. 4, pp. 2849–2865, April 2019.

[19] A. Tang, S. Roy, and X. Wang, "Coded caching for wireless backhaul networks with unequal link rates," *IEEE Transactions on Communications*, vol. 66, no. 1, pp. 1–13, Jan 2018.

[20] A. M. Daniel and W. Yu, "Optimization of heterogeneous coded caching," *IEEE Transactions on Information Theory*, pp. 1–1, 2019.

[21] A. M. Ibrahim, A. A. Zewail, and A. Yener, "Coded caching for heterogeneous systems: An optimization perspective," *IEEE Transactions on Communications*, vol. 67, no. 8, pp. 5321–5335, Aug 2019.

[22] ——, "Device-to-device coded-caching with distinct cache sizes," *IEEE Transactions on Communications*, pp. 1–1, 2020.

[23] C. Chang and C. Wang, "Coded caching with full heterogeneity: Exact capacity of the two-user/two-file case," in *2019 IEEE International Symposium on Information Theory (ISIT)*, July 2019, pp. 6–10.

[24] Y. Lu, W. Chen, and H. V. Poor, "Coded caching under heterogeneous user preferences: An effective throughput perspective," in *ICC 2019 - 2019 IEEE International Conference on Communications (ICC)*, May 2019, pp. 1–6.

[25] J. Hachem, N. Karamchandani, and S. N. Diggavi, "Coded caching for multi-level popularity and access," *IEEE Transactions on Information Theory*, vol. 63, no. 5, pp. 3108–3141, May 2017.

[26] J. Hachem, N. Karamchandani, and S. Diggavi, "Effect of number of users in multi-level coded caching," in *2015 IEEE International Symposium on Information Theory (ISIT)*, June 2015, pp. 1701–1705.

[27] T. Li, M. Ashraphijuo, X. Wang, and P. Fan, "Traffic off-loading with energy-harvesting small cells and coded content caching," *IEEE Transactions on Communications*, vol. 65, no. 2, pp. 906–917, Feb 2017.

[28] M. Ji, A. M. Tulino, J. Llorca, and G. Caire, "Order-optimal rate of caching and coded multicasting with random demands," *IEEE Transactions on Information Theory*, vol. 63, no. 6, pp. 3923–3949, June 2017.

[29] 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.

[30] S. A. Saberali, L. Lampe, and I. F. Blake, "Full characterization of optimal uncoded placement for the structured clique cover delivery of nonuniform demands," *IEEE Transactions on Information Theory*, vol. 66, no. 1, pp. 633–648, Jan 2020.

[31] U. Niesen and M. A. Maddah-Ali, "Coded caching with nonuniform demands," *IEEE Transactions on Information Theory*, vol. 63, no. 2, pp. 1146–1158, Feb 2017.

[32] 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, 2018.

[33] 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 Transactions on Information Theory*, vol. 65, no. 12, pp. 7756–7770, Dec 2019.

[34] M. Ji, K. Shanmugam, G. Vettigli, J. Llorca, A. M. Tulino, and G. Caire, "An efficient multiple-groupcast coded multicasting scheme for finite fractional caching," in *2015 IEEE International Conference on Communications (ICC)*, June 2015, pp. 3801–3806.

[35] Q. Yang and D. Gündüz, "Coded caching and content delivery with heterogeneous distortion requirements," *IEEE Transactions on Information Theory*, vol. 64, no. 6, pp. 4347–4364, June 2018.

[36] A. M. Ibrahim, A. A. Zewail, and A. Yener, "On coded caching with heterogeneous distortion requirements," in *2018 Information Theory and Applications Workshop (ITA)*, Feb 2018, pp. 1–9.

[37] G. Ma, Z. Wang, M. Zhang, J. Ye, M. Chen, and W. Zhu, "Understanding performance of edge content caching for mobile video streaming," *IEEE Journal on Selected Areas in Communications*, vol. 35, no. 5, pp. 1076–1089, 2017.

[38] Y. Lu, C. Li, W. Chen, and H. Vincent Poor, "On the effective throughput of coded caching with heterogeneous user preferences: A game theoretic perspective," *IEEE Transactions on Communications*, vol. 69, no. 3, pp. 1387–1402, 2021.

[39] G. Dán and N. Carlsson, "Dynamic content allocation for cloud-assisted service of periodic workloads," in *IEEE INFOCOM 2014 - IEEE Conference on Computer Communications*, 2014, pp. 853–861.

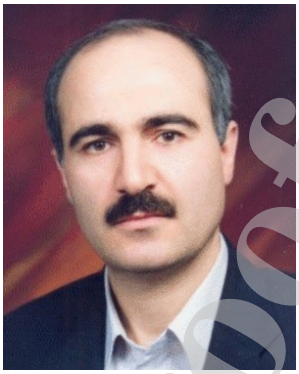
Abdollah Ghaffari Sheshjavani received the B.Sc. and M.Sc. degrees in computer engineering from Shahid Sattari Aeronautical University of Science and Technology and Tarbiat Modares University, Tehran, Iran, in 2008 and 2013, respectively. He is currently pursuing the Ph.D. degree with the Department of Electrical and Computer Engineering, University of Tehran, Iran. His research interests include simulation and data analysis, wired/wireless networks, P2P networks, video streaming, distributed systems, and content caching in telecommunication networks.

Ahmad Khonsari received the B.Sc. degree in electrical and computer engineering from Shahid Beheshti University, Iran, and M.Sc. degree in computer engineering from the Iran University of Science and Technology (IUST), Iran, and Ph.D. degree in computer science from the University of Glasgow, U.K., in 1991, 1996, and 2003, respectively. He is currently an Associate Professor with the Department of Electrical and Computer Engineering, University of Tehran, Iran, and a Researcher with the School of Computer Science, Institute for Research in Fundamental Sciences (IPM), Iran. His research interests include simulation and data analysis, performance modeling/evaluation, wired/wireless networks, cloud and distributed systems, and high performance computer architectures.

Seyed Pooya Shariatpanahi received the B.Sc., M.Sc., and Ph.D. degrees from the Department of Electrical Engineering, Sharif University of Technology, Tehran, Iran, in 2006, 2008, and 2013, respectively. He is currently an Assistant Professor with the School of Electrical and Computer Engineering, University of Tehran. Before joining the University of Tehran, he was a Researcher with the Institute for Research in Fundamental Sciences (IPM), Tehran. His research interests include information theory, network science, wireless communications, and complex systems. He was a recipient of the Gold Medal at the National Physics Olympiad, in 2001.

Masoumeh Moradian received the B.S., M.S., and Ph.D. degrees from the Sharif University of Technology, Tehran, Iran, in 2007, 2010, and 2016, respectively, all in electrical engineering. She was a Visiting Scholar with the Chinese University of Hong Kong in 2015. She is currently a Postdoctoral Researcher with the School of Computer Science, Institute for Research in Fundamental Sciences (IPM), Tehran, Iran. Her current research interests include energy harvesting communication networks, queueing theory and network stochastic optimization

Abdollah Ghaffari Sheshjavani



Ahmad Khonsari

Seyed Pooya Shariatpanahi

Masoumeh Moradian

Declaration of interests

☒ The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

☐The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: