Centralized Coded Caching with Heterogeneous Cache Sizes

Abdelrahman M. Ibrahim, Ahmed A. Zewail, and Aylin Yener

Wireless Communications and Networking Laboratory (WCAN)
Electrical Engineering and Computer Science Department
The Pennsylvania State University, University Park, PA 16802.
{*ami137,zewail*}*@psu.edu* *yener@engr.psu.edu*

***Abstract*—Coded caching can improve fundamental limits of communication, utilizing storage memory at individual users. This paper considers a centralized coded caching system, introducing heterogeneous cache sizes at the users, i.e., the users' cache memories are of different size. The goal is to design cache placement and delivery policies that minimize the worst-case delivery load on the server. To that end, the paper proposes an optimization framework for cache placement and delivery schemes which explicitly accounts for the heterogeneity of the cache sizes. We also characterize explicitly the optimal caching scheme, for the case where the sum of the users' cache sizes is smaller than or equal to the library size.**

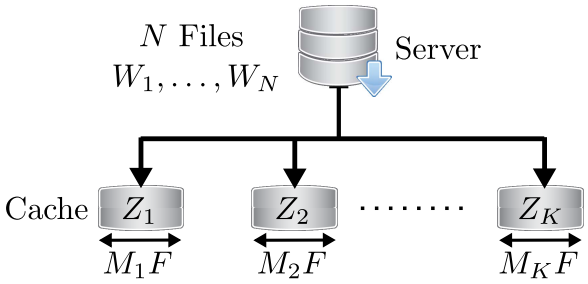

Fig. 1: Caching system with heterogeneous cache sizes.

I. Introduction

Caching [1]–[9] aims to alleviate network congestion during peak-traffic hours, by pushing data into the cache memories at the network edge during off-peak hours. The former is often called the delivery phase and the latter the placement phase. The contents requested by the users, are thus partially available at the users' local cache memories and the remaining pieces need to be delivered, e.g., by a server. Reference [1] has introduced the fundamental concept of *coded caching*, where the placement and delivery phases are jointly optimized to minimize the delivery load, by creating multicast opportunities.

Extensive effort has transpired towards studying fundamental limits of coded caching in various setups since the introduction of coded caching systems with equal cache sizes in [1]. In particular, reference [2] has considered a decentralized system where the users populate their cache memories independently, i.e., the placement phase is decentralized. Despite lack of coordination, the server is able to create multicast transmissions and the resulting delivery load is close to the centralized delivery load in [1] for large networks. Several other network architectures with caching capabilities have also been considered, such as two-hop networks [5], device-to-device (D2D) networks [6], and interference networks [7], [8].

Today's networks contain variety of devices at end users, e.g., laptops, smart phones, etc., with varying storage capabilities, which in turn motivates considering the realistic model where the end users have heterogeneous cache sizes. Reference [3] has extended the *decentralized* scheme in [2] to systems with heterogeneous cache sizes. Reference [4] has proposed a group-based delivery scheme that achieves lower delivery load for systems where the number of users is greater than the number of files. For centralized caching however, the scheme in [1] cannot be directly extended to the setup with heterogeneous cache sizes. Thus, a new scheme is needed.

In this paper, we tackle this challenge and propose a centralized coded caching scheme in systems with heterogeneous cache sizes. We formulate and solve the optimization problem over the parameters of the cache placement and delivery schemes, in order to minimize the worst-case delivery load. We consider the set of feasible placement policies, $\mathfrak{A}$, under the uncoded prefetching assumption [9], i.e., the users cache uncoded pieces of the files. For given cache contents, we consider the set of delivery schemes, $\mathfrak{D}$, that includes all possible unicast/multicast transmissions, and guarantees that the users' requests will be satisfied. For the case where library size is larger than or equal to accumulated users' cache sizes, we derive the explicit caching scheme that yields the optimal worst-case delivery load under uncoded prefetching $R^*_{\mathfrak{A}}$. Finally, we compare the delivery load of our scheme with the lower bounds in [3], [4].

Notation: Vectors are represented by boldface letters, sets of policies are represented by calligraphic letters, e.g., $\mathfrak{A}$, $\oplus$ refers to bitwise XOR operation, $|W|$ denotes cardinality of W, $\{\}$ denotes the empty set, $[K] := \{1, \ldots, K\}$, and $2^{[K]}$ denotes the power set of $[K]$.

II. System model

We consider the caching system shown in Fig. 1, where a single server is connected to K users via a shared error-free multicast link [1]. A library $\{W_1, \ldots, W_N\}$ of N files, each with size F bits, is stored at the server. We consider a

978-1-5090-4183-1/17/$31.00 ©2017 IEEE

heterogeneous system, where user k is equipped with a cache memory of size M_kF bits, i.e., M_k is the memory size of user k normalized by the file size. Without loss of generality, we assume that $M_1 \leq M_2 \leq \cdots \leq M_K$. Additionally, we define $m_k = M_k/N$ to denote the memory size of user k normalized by the library size NF, i.e., $m_k \in [0,1]$ for $M_k \in [0,N]$. The cache size vector is denoted by $\boldsymbol{M} = [M_1, \ldots, M_K]$ and its normalized version by the library size is denoted by $\boldsymbol{m} = [m_1, \ldots, m_K]$. We focus on the case, where the number of files is at least as many as the number of users, i.e., $N \geq K$.

The system operates over two phases: placement phase and delivery phase. In the placement phase, the server populates the users' cache memories without the knowledge of users' demands that will be made in the delivery phase. In particular, user k stores a subset Z_k of the files library, subject to its cache size constraint. In the delivery phase, user k requests a file W_{d_k} from the server. The users' demands are uniform and independent, i.e., the demand vector $\boldsymbol{d} = [d_1, \ldots, d_K]$ consists of identical and independent uniform random variables over the files as in [1]. In order to serve the users' demands, the server transmits a sequence of unicast/multicast signals, $X_{\mathcal{T},\boldsymbol{d}}$, to the users in $\mathcal{T} \in 2^{[K]} - \{\}$, which represent all possible transmission sets. Define the transmission variable $v_{\mathcal{T}} \in [0,1]$ which represents the fraction of the file delivered to the users in $\mathcal{T}$ via $X_{\mathcal{T},\boldsymbol{d}}$. Thus, $X_{\mathcal{T},\boldsymbol{d}}$ has length $v_{\mathcal{T}}F$ bits. At the end of the delivery phase, user k must be able to decode $\hat{W}_{d_k}$ reliably. Formally, we have the following definition.

Definition 1. *For a given normalized cache size vector $\boldsymbol{m}$, the delivery load $R(\boldsymbol{m})$ is said to be achievable if for every $\epsilon > 0$ and large enough F, there exists a caching scheme such that $\max_{\boldsymbol{d},k\in[K]} Pr(\hat{W}_{d_k} \neq W_{d_k}) \leq \epsilon$. Moreover, the infimum over all achievable delivery loads is denoted by $R^*(\boldsymbol{m})$.* ■

In this work, we focus on uncoded prefetching where the users cache uncoded bits [9]. We also have that user k divides its cache size M_kF equally over the files, since the demand is uniform, dedicating m_kF bits to each file. We denote this class of cache placement policies by $\mathfrak{A}$. We denote the set of all delivery policies by $\mathfrak{D}$.

Definition 2. *Under a placement policy in $\mathfrak{A}$, and any delivery policy in $\mathfrak{D}$, the worst-case delivery load is defined as $R_{\mathfrak{A},\mathfrak{D}} := \max_{\boldsymbol{d}} R_{\boldsymbol{d},\mathfrak{A},\mathfrak{D}} = \sum_{\mathcal{T}\in 2^{[K]}-\{\}} v_{\mathcal{T}}$. Moreover, by taking the infimum over all caching schemes, we get $R^*_{\mathfrak{A}}$.* ■

III. Caching Model: The Three-User Case

For clarity of exposition, we first develop cache placement and delivery schemes for three users, i.e., $K = 3$. We then generalize to $K > 3$.

A. Placement Phase

Each file W_l is partitioned into 2^3 disjoint subfiles, which are labeled by the users storing them, as shown in Fig. 2. $\tilde{W}_{l,\mathcal{S}}$ denotes the subfile stored at the users in $\mathcal{S}$. For example, $\tilde{W}_{l,\{1,2\}}$ is stored at users 1 and 2, and $\tilde{W}_{l,\{\}}$ is not stored at any user. The set of all possible allocation sets is the power set of $\{1,2,3\}$. We assume that subfile sizes are the same for all

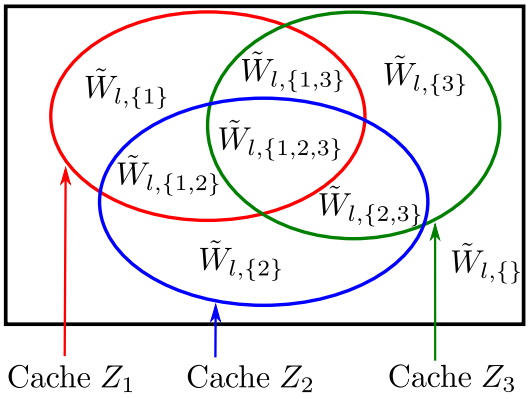

Fig. 2: The partitions of file W_l over the storage sets.

files, i.e., $|\tilde{W}_{l,\mathcal{S}}| = a_{\mathcal{S}}F$ bits, $\forall l \in [N]$, where the allocation variable $a_{\mathcal{S}} \in [0,1]$ is the fraction of the file stored at $\mathcal{S}$. Hence, the cache content of user k is given by

$$Z_k = \bigcup_{l\in[N]} \bigcup_{\mathcal{S}\in 2^{[K]} : k\in\mathcal{S}} \tilde{W}_{l,\mathcal{S}}.$$

Moreover, the sets $\mathcal{S} \in 2^{[K]}$ define a partition of each file W_l, see Fig. 2. We have

$$a_{\{\}}+a_{\{1\}}+a_{\{2\}}+a_{\{3\}}+a_{\{1,2\}}+a_{\{1,3\}}+a_{\{2,3\}}+a_{\{1,2,3\}}=1. \quad (1)$$

The fact that user k dedicates m_kF bits of its cache memory to each file, implies that the cache sizes constraints can be expressed as follows

$$a_{\{1\}} + a_{\{1,2\}} + a_{\{1,3\}} + a_{\{1,2,3\}} \leq m_1, \quad (2)$$
$$a_{\{2\}} + a_{\{1,2\}} + a_{\{2,3\}} + a_{\{1,2,3\}} \leq m_2, \quad (3)$$
$$a_{\{3\}} + a_{\{1,3\}} + a_{\{2,3\}} + a_{\{1,2,3\}} \leq m_3. \quad (4)$$

Consequently, for $K = 3$, the set of feasible cache placement schemes is characterized by $a_{\mathcal{S}}, \mathcal{S} \in 2^{[K]}$ that satisfy (1)-(4).

B. Delivery Phase

Recall from Section II that $X_{\mathcal{T},\boldsymbol{d}}$ is transmitted to the users in $\mathcal{T}$. At the end of transmission, user k must be able to reconstruct W_{d_k} from $X_{\mathcal{T},\boldsymbol{d}}$, $k \in \mathcal{T}$ and its cache content Z_k.

1) Structure of $X_{\mathcal{T},\boldsymbol{d}}$: The multicast signals are formed by XORing file pieces of equal size. For example, the multicast signal $X_{\{1,2\},\boldsymbol{d}} = W_{d_1}^{\{1,2\}} \oplus W_{d_2}^{\{1,2\}}$, where $W_{d_j}^{\{1,2\}} \subset W_{d_j}$, and $|W_{d_1}^{\{1,2\}}| = |W_{d_2}^{\{1,2\}}| = v_{\{1,2\}}F$ bits. Similarly, we have

$$X_{\{1,3\},\boldsymbol{d}} = W_{d_1}^{\{1,3\}} \oplus W_{d_3}^{\{1,3\}},$$
$$X_{\{2,3\},\boldsymbol{d}} = W_{d_2}^{\{2,3\}} \oplus W_{d_3}^{\{2,3\}},$$
$$X_{\{1,2,3\},\boldsymbol{d}} = W_{d_1}^{\{1,2,3\}} \oplus W_{d_2}^{\{1,2,3\}} \oplus W_{d_3}^{\{1,2,3\}}.$$

Additionally, a unicast transmission delivers the fraction of the requested file that is not available at the user's local cache and will not be delivered by the multicast transmissions. For example, the unicast transmission to user 1 is given by

$$X_{\{1\},\boldsymbol{d}} = W_{d_1} - \bigcup_{\mathcal{S}:1\in\mathcal{S}} \tilde{W}_{d_1,\mathcal{S}} - W_{d_1}^{\{1,2\}} \bigcup W_{d_1}^{\{1,3\}} \bigcup W_{d_1}^{\{1,2,3\}},$$

where $\bigcup_{\mathcal{S}:1\in\mathcal{S}} \tilde{W}_{d_1,\mathcal{S}}$ is the fraction of the requested file W_{d_1} available locally at user 1 and $W_{d_1}^{\{1,2\}} \bigcup W_{d_1}^{\{1,3\}} \bigcup W_{d_1}^{\{1,2,3\}}$ is delivered to user 1 via multicast transmissions. Similarly, the remaining unicast signals are given by

$$X_{\{2\},\boldsymbol{d}} = W_{d_2} - \bigcup_{\mathcal{S}:2\in\mathcal{S}} \tilde{W}_{d_2,\mathcal{S}} - W_{d_2}^{\{1,2\}} \bigcup W_{d_2}^{\{2,3\}} \bigcup W_{d_2}^{\{1,2,3\}},$$
$$X_{\{3\},\boldsymbol{d}} = W_{d_3} - \bigcup_{\mathcal{S}:3\in\mathcal{S}} \tilde{W}_{d_3,\mathcal{S}} - W_{d_3}^{\{1,3\}} \bigcup W_{d_3}^{\{2,3\}} \bigcup W_{d_3}^{\{1,2,3\}}.$$

2) Construction of $W_{d_j}^{\mathcal{T}}$: In order to ensure that each user in $\mathcal{T}-\{j\}$ is able to cancel $W_{d_j}^{\mathcal{T}}$ from $X_{\mathcal{T},\boldsymbol{d}}$, $W_{d_j}^{\mathcal{T}}$ is constructed from the subfiles cached by all the users in $\mathcal{T}-\{j\}$, i.e., subfiles $\tilde{W}_{d_j,\mathcal{S}}$ where $\mathcal{T}-\{j\}\subset\mathcal{S}$ and $j\notin\mathcal{S}$. In particular, $W_{d_j}^{\mathcal{T}}$ is formed from the pieces $W_{d_j,\mathcal{S}}^{\mathcal{T}}$, i.e., the subsets of $W_{d_j}^{\mathcal{T}}$ that are stored at the users in $\mathcal{S}$. For example, the pieces composing $X_{\{1,2\}}$ are defined by

$$W_{d_1}^{\{1,2\}} = W_{d_1,\{2\}}^{\{1,2\}} \bigcup W_{d_1,\{2,3\}}^{\{1,2\}}, W_{d_2}^{\{1,2\}} = W_{d_2,\{1\}}^{\{1,2\}} \bigcup W_{d_2,\{1,3\}}^{\{1,2\}}.$$

Similarly, the remaining pieces are defined by

$$W_{d_1}^{\{1,3\}} = W_{d_1,\{3\}}^{\{1,3\}} \bigcup W_{d_1,\{2,3\}}^{\{1,3\}},\ W_{d_1}^{\{1,2,3\}} = W_{d_1,\{2,3\}}^{\{1,2,3\}},$$
$$W_{d_3}^{\{1,3\}} = W_{d_3,\{1\}}^{\{1,3\}} \bigcup W_{d_3,\{1,2\}}^{\{1,3\}},\ W_{d_2}^{\{1,2,3\}} = W_{d_2,\{1,3\}}^{\{1,2,3\}},$$
$$W_{d_2}^{\{2,3\}} = W_{d_2,\{3\}}^{\{2,3\}} \bigcup W_{d_2,\{1,3\}}^{\{2,3\}},\ W_{d_3}^{\{1,2,3\}} = W_{d_3,\{1,2\}}^{\{1,2,3\}},$$
$$W_{d_3}^{\{2,3\}} = W_{d_3,\{2\}}^{\{2,3\}} \bigcup W_{d_3,\{1,2\}}^{\{2,3\}}.$$

Define the assignment variable $u_{\mathcal{S}}^{\mathcal{T}} \in [0, a_{\mathcal{S}}]$ to represent the fraction of $\tilde{W}_{d_j,\mathcal{S}}$ involved in $X_{\mathcal{T},\boldsymbol{d}}$, i.e., $|W_{d_j,\mathcal{S}}^{\mathcal{T}}| = u_{\mathcal{S}}^{\mathcal{T}} F$ bits. Thus, each subfile $\tilde{W}_{d_j,\mathcal{S}}$ is partitioned into $W_{d_j,\mathcal{S}}^{\mathcal{T}}$, and $W_{d_j,\mathcal{S}}^{\{\}}$ which denotes the remaining piece. We have

$$\tilde{W}_{d_3,\{1,2\}} = W_{d_3,\{1,2\}}^{\{1,3\}} \bigcup W_{d_3,\{1,2\}}^{\{2,3\}} \bigcup W_{d_3,\{1,2\}}^{\{1,2,3\}} \bigcup W_{d_3,\{1,2\}}^{\{\}},$$
$$\tilde{W}_{d_2,\{1,3\}} = W_{d_2,\{1,3\}}^{\{1,2\}} \bigcup W_{d_2,\{1,3\}}^{\{2,3\}} \bigcup W_{d_2,\{1,3\}}^{\{1,2,3\}} \bigcup W_{d_2,\{1,3\}}^{\{\}},$$
$$\tilde{W}_{d_1,\{2,3\}} = W_{d_1,\{2,3\}}^{\{1,2\}} \bigcup W_{d_1,\{2,3\}}^{\{2,3\}} \bigcup W_{d_1,\{2,3\}}^{\{1,2,3\}} \bigcup W_{d_1,\{2,3\}}^{\{\}}.$$

3) Delivery phase constraints: Recall that we have defined the transmission variable, $v_{\mathcal{T}} \in [0,\ 1]$ as $|X_{\mathcal{T},\boldsymbol{d}}|/F$ in Section II. We now describe the delivery scheme by constraints on the transmission variables $v_{\mathcal{T}}$, and the assignment variables $u_{\mathcal{S}}^{\mathcal{T}}$ defined above in 2). First, in order to satisfy the users' demands, the signals sent by the server must *complete* the file requested by each user, which is represented by the following delivery constraints

$$v_{\{1\}} + v_{\{1,2\}} + v_{\{1,3\}} + v_{\{1,2,3\}} \geq 1 - m_1, \quad (5)$$
$$v_{\{2\}} + v_{\{1,2\}} + v_{\{2,3\}} + v_{\{1,2,3\}} \geq 1 - m_2, \quad (6)$$
$$v_{\{3\}} + v_{\{1,3\}} + v_{\{2,3\}} + v_{\{1,2,3\}} \geq 1 - m_3. \quad (7)$$

That is, m_k represents the local caching gain. Moreover, the structure of the transmitted pieces imposes the following constraints

$$v_{\{1,2\}} = u_{\{2\}}^{\{1,2\}} + u_{\{2,3\}}^{\{1,2\}} = u_{\{1\}}^{\{1,2\}} + u_{\{1,3\}}^{\{1,2\}}, \quad (8)$$
$$v_{\{1,3\}} = u_{\{3\}}^{\{1,3\}} + u_{\{2,3\}}^{\{1,3\}} = u_{\{1\}}^{\{1,3\}} + u_{\{1,2\}}^{\{1,3\}}, \quad (9)$$

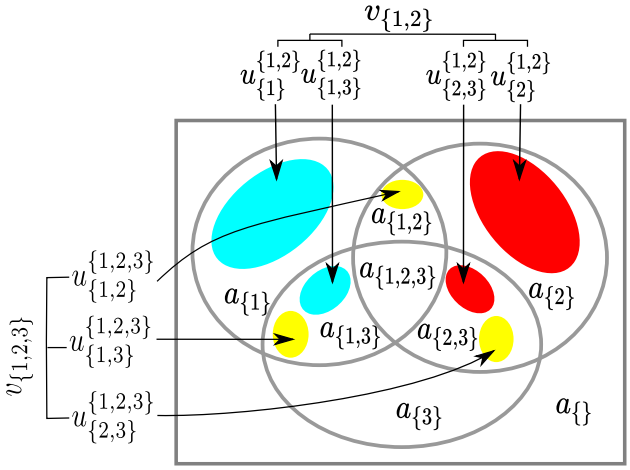

Fig. 3: The relation between transmission, allocation, and assignment variables.

$$v_{\{2,3\}} = u_{\{3\}}^{\{2,3\}} + u_{\{1,3\}}^{\{2,3\}} = u_{\{2\}}^{\{2,3\}} + u_{\{1,2\}}^{\{2,3\}}, \quad (10)$$
$$v_{\{1,2,3\}} = u_{\{2,3\}}^{\{1,2,3\}} = u_{\{1,3\}}^{\{1,2,3\}} = u_{\{1,2\}}^{\{1,2,3\}}. \quad (11)$$

The relationship between $v_{\mathcal{T}}$, the allocation variables, $a_{\mathcal{S}}$, and the assignment variables, $u_{\mathcal{S}}^{\mathcal{T}}$, is further illustrated in Fig. 3. We observe that the transmission variable $v_{\{1,2\}}$ is limited by the amount of side information stored at user 1 and not available at user 2, i.e., $a_{\{1\}} + a_{\{1,3\}}$, and the amount of side information stored at user 2 and not available at user 1, i.e., $a_{\{2\}} + a_{\{2,3\}}$. Additionally, $v_{\{1,2\}}$ and $v_{\{1,2,3\}}$ are both limited by $a_{\{1,3\}}$ and $a_{\{2,3\}}$. Hence, we have the following side information constraints

$$v_{\{1,2\}} + v_{\{1,2,3\}} \leq a_{\{1\}} + a_{\{1,3\}}, \quad (12)$$
$$v_{\{1,2\}} + v_{\{1,2,3\}} \leq a_{\{2\}} + a_{\{2,3\}}. \quad (13)$$

Similarly, the other side information constraints are given by

$$v_{\{1,3\}} + v_{\{1,2,3\}} \leq a_{\{1\}} + a_{\{1,2\}}, \quad (14)$$
$$v_{\{1,3\}} + v_{\{1,2,3\}} \leq a_{\{3\}} + a_{\{2,3\}}, \quad (15)$$
$$v_{\{2,3\}} + v_{\{1,2,3\}} \leq a_{\{2\}} + a_{\{1,2\}}, \quad (16)$$
$$v_{\{2,3\}} + v_{\{1,2,3\}} \leq a_{\{3\}} + a_{\{1,3\}}. \quad (17)$$

Moreover, subfile $\tilde{W}_{d_3,\{1,2\}}$ can be utilized in the multicast transmissions that include user 3, i.e., $X_{\{1,3\},\boldsymbol{d}}$, $X_{\{2,3\},\boldsymbol{d}}$, $X_{\{1,2,3\},\boldsymbol{d}}$. The following constraint prevents transmitting redundant bits to user 3

$$u_{\{1,2\}}^{\{1,3\}} + u_{\{1,2\}}^{\{2,3\}} + u_{\{1,2\}}^{\{1,2,3\}} \leq a_{\{1,2\}}. \quad (18)$$

Similarly, the constraints that prevent transmitting redundant bits to users 1 and 2 are given by

$$u_{\{1,3\}}^{\{1,2\}} + u_{\{1,3\}}^{\{2,3\}} + u_{\{1,3\}}^{\{1,2,3\}} \leq a_{\{1,3\}}, \quad (19)$$
$$u_{\{2,3\}}^{\{1,2\}} + u_{\{2,3\}}^{\{1,3\}} + u_{\{2,3\}}^{\{1,2,3\}} \leq a_{\{2,3\}}. \quad (20)$$

In summary, given the allocation variables, the set of feasible delivery schemes, i.e., schemes that satisfy all users' requests,

in a three-user system, is characterized by (5)-(20), $0 \leq u_{\mathcal{S}}^{\mathcal{T}} \leq a_{\mathcal{S}}$, and $0 \leq v_{\mathcal{T}} \leq 1$.

IV. Caching Model: The K-user case

In this section, we generalize the cache placement and delivery schemes to systems with K users.

A. Placement phase

In the general case, each file W_l is partitioned into 2^K subfiles, and the set of feasible cache placement policies for a given $\boldsymbol{m}$, $\mathfrak{A}(\boldsymbol{m})$, is defined as

$$\left\{ \boldsymbol{a} \in [0,1]^{2^K} \middle| \sum_{\mathcal{S} \in 2^{[K]}} a_{\mathcal{S}} = 1, \sum_{\mathcal{S} \in 2^{[K]} \,:\, k \in \mathcal{S}} a_{\mathcal{S}} \leq m_k, \forall k \in [K] \right\}, \quad (21)$$

where the allocation vector $\boldsymbol{a}$ represents the collection of allocation variables $a_{\mathcal{S}}$, $\mathcal{S} \in 2^{[K]}$. The cache placement procedure is summarized in Algorithm 1, for a given allocation vector $\boldsymbol{a} \in \mathfrak{A}(\boldsymbol{m})$.

Algorithm 1 Cache placement procedure

Input: $\{W_1, \ldots, W_N\}$ and $\boldsymbol{a}$
Output: $Z_k, k \in [K]$

for $l \in [N]$ **do**
  Partition file W_l into subfiles $\tilde{W}_{l,\mathcal{S}}, \mathcal{S} \in 2^{[K]}$ such that $|\tilde{W}_{l,\mathcal{S}}| = a_{\mathcal{S}} F$.
end for
for $k \in [K]$ **do**
  $Z_k \leftarrow \bigcup_{l \in [N]} \bigcup_{\mathcal{S} \in 2^{[K]} \,:\, k \in \mathcal{S}} \tilde{W}_{l,\mathcal{S}}$
end for

B. Delivery phase

In general, $X_{\mathcal{T},\boldsymbol{d}} = \oplus_{j \in \mathcal{T}} W_{d_j}^{\mathcal{T}}$, where $|W_{d_j}^{\mathcal{T}}| = v_{\mathcal{T}} F$ bits, $\forall j \in \mathcal{T}$. Additionally, a multicast transmission to the users in $\mathcal{T}$ is constrained by the side information stored at the sets

$$\mathcal{B}_j^{\mathcal{T}} := \left\{ \mathcal{S} \in 2^{[K]} : \mathcal{T} - \{j\} \subset \mathcal{S}, j \notin \mathcal{S} \right\}, \forall j \in \mathcal{T}, \quad (22)$$

where $\mathcal{B}_j^{\mathcal{T}}$ represents the sets containing the side information stored at $\mathcal{T} - \{j\}$ and not available at user j. The set of all storage sets related to $\mathcal{T}$ is denoted by

$$\mathcal{B}^{\mathcal{T}} := \bigcup_{j \in \mathcal{T}} \mathcal{B}_j^{\mathcal{T}}. \quad (23)$$

Moreover, for $|\mathcal{T}| \geq 2$, each piece $W_{d_j}^{\mathcal{T}}$ is partitioned over the sets $\mathcal{S} \in \mathcal{B}_j^{\mathcal{T}}$, i.e.,

$$W_{d_j}^{\mathcal{T}} = \bigcup_{\mathcal{S} \in \mathcal{B}_j^{\mathcal{T}}} W_{d_j,\mathcal{S}}^{\mathcal{T}}, \quad (24)$$

where $|W_{d_j,\mathcal{S}}^{\mathcal{T}}| = u_{\mathcal{S}}^{\mathcal{T}} F$ bits.

For given $\boldsymbol{m}$ and $\boldsymbol{a}$, the set of feasible delivery schemes is denoted by $\mathfrak{D}(\boldsymbol{m}, \boldsymbol{a})$, and characterized by the transmission and assignment variables that satisfy the following conditions

$$\sum_{\mathcal{S} \in \mathcal{B}_j^{\mathcal{T}}} u_{\mathcal{S}}^{\mathcal{T}} = v_{\mathcal{T}}, \ \forall \mathcal{T} \in 2^{[K]} - \{\}, \ \forall j \in \mathcal{T}, \quad (25)$$

Algorithm 2 Delivery procedure

Input: $\boldsymbol{d}, \boldsymbol{a}, \boldsymbol{u}, \boldsymbol{v}$, and $\tilde{W}_{l,\mathcal{S}}, \mathcal{S} \in 2^{[K]}, l \in [N]$
Output: $X_{\mathcal{T},\boldsymbol{d}}, \mathcal{T} \in 2^{[K]} - \{\}$

{Partitioning}
for $\{\mathcal{S} \in 2^{[K]} : 1 \leq |\mathcal{S}| \leq K-1\}$ **do**
  for $\{j \in [K] : j \notin \mathcal{S}\}$ **do**
    Partition $\tilde{W}_{d_j,\mathcal{S}}$ into $W_{d_j,\mathcal{S}}^{\mathcal{T}}$, $\left\{\mathcal{T} \in 2^{[K]} : j \in \mathcal{T}, \mathcal{T} \cap \mathcal{S} \neq \{\}\right\}$, such that $|W_{d_j,\mathcal{S}}^{\mathcal{T}}| = u_{\mathcal{S}}^{\mathcal{T}} F$ and $W_{d_j,\mathcal{S}}^{\{\}}$ is the remaining segment.
  end for
end for
{Delivery scheme}
for $\mathcal{T} \in 2^{[K]} - \{\}$ **do**
  if $\mathcal{T} = \{j\}$ **then**
    $W_{d_j}^{\{j\}} := W_{d_j} - \bigcup_{\mathcal{S}: j \in \mathcal{S}} \tilde{W}_{d_j,\mathcal{S}} - \bigcup_{\mathcal{T}'} \bigcup_{\mathcal{S}} W_{d_j,\mathcal{S}}^{\mathcal{T}'}$
    $X_{\{j\},\boldsymbol{d}} \leftarrow W_{d_j}^{\{j\}}$ {Unicast transmissions}
  else
    $W_{d_j}^{\mathcal{T}} := \bigcup_{\mathcal{S} \in \mathcal{B}_j^{\mathcal{T}}} W_{d_j,\mathcal{S}}^{\mathcal{T}}$
    $X_{\mathcal{T},\boldsymbol{d}} \leftarrow \oplus_{j \in \mathcal{T}} W_{d_j}^{\mathcal{T}}$ {Multicast transmissions}
  end if
end for

$$\sum_{\mathcal{T} \in 2^{[K]} - \{\} \,:\, k \in \mathcal{T}} v_{\mathcal{T}} \geq 1 - m_k, \ \forall k \in [K], \quad (26)$$

$$\sum_{\mathcal{T} \in 2^{[K]} - \{\} \,:\, \{j\} \cup \mathcal{S}' \subset \mathcal{T}} v_{\mathcal{T}} \leq \sum_{\mathcal{S} \in 2^{[K]} \,:\, \mathcal{S}' \subset \mathcal{S}, j \notin \mathcal{S}} a_{\mathcal{S}}, \ \forall j \notin \mathcal{S}',$$
$$\forall \mathcal{S}' \in \left\{ \tilde{\mathcal{S}} \in 2^{[K]} : 1 \leq |\tilde{\mathcal{S}}| \leq K-2 \right\}, \quad (27)$$

$$\sum_{\mathcal{T} \in 2^{[K]} - \{\} \,:\, j \in \mathcal{T}, \mathcal{T} \cap \mathcal{S} \neq \{\}, |\mathcal{T}| \leq |\mathcal{S}| + 1} u_{\mathcal{S}}^{\mathcal{T}} \leq a_{\mathcal{S}}, \ \forall j \notin \mathcal{S},$$
$$\forall \mathcal{S} \in \left\{ \tilde{\mathcal{S}} \in 2^{[K]} : \ 2 \leq |\tilde{\mathcal{S}}| \leq K-1 \right\}, \quad (28)$$

$$0 \leq u_{\mathcal{S}}^{\mathcal{T}} \leq a_{\mathcal{S}}, \ \forall \mathcal{T} \in 2^{[K]} - \{\}, \ \forall \mathcal{S} \in \mathcal{B}^{\mathcal{T}}, \quad (29)$$

$$0 \leq v_{\mathcal{T}} \leq 1, \ \forall \mathcal{T} \in 2^{[K]} - \{\}. \quad (30)$$

The delivery scheme is summarized in Algorithm 2, for given allocation ($\boldsymbol{a}$), assignment ($\boldsymbol{u}$), and transmission vectors ($\boldsymbol{v}$).

V. Optimization and Performance

In Sections III-A and IV-A, we have illustrated that a cache placement policy is completely characterized by the allocation vector $\boldsymbol{a}$, which represents the fraction of files stored exclusively at each subset of users $\mathcal{S} \subset [K]$. Sections III-B and IV-B have demonstrated that a delivery scheme is characterized by the pair $(\boldsymbol{u}, \boldsymbol{v})$, where the size and structure of the transmitted signals are represented by the transmission vector $\boldsymbol{v}$ and the assignment vector $\boldsymbol{u}$, respectively. Next, we formulate an optimization problem to minimize the worst-case delivery load, i.e., the sum of transmission variables, by jointly optimizing the cache placement and delivery schemes.

A. Optimizing the Caching Scheme

For a given normalized memory vector $\boldsymbol{m}$, the following optimization problem characterizes the minimum worst-case

delivery load $R^*_{\mathfrak{A},\mathfrak{D}}(\boldsymbol{m})$ and the optimal caching scheme, i.e., the optimal values for $\boldsymbol{a}$, $\boldsymbol{v}$, and $\boldsymbol{u}$.

$$\textbf{\textit{O1}}: \min_{\boldsymbol{a},\boldsymbol{u},\boldsymbol{v}} \quad \sum_{\mathcal{T}\in 2^{[K]}-\{\}} v_{\mathcal{T}} \tag{31a}$$

$$\text{subject to} \quad \boldsymbol{a}\in\mathfrak{A}(\boldsymbol{m}), \tag{31b}$$

$$(\boldsymbol{u},\boldsymbol{v})\in\mathfrak{D}(\boldsymbol{m},\boldsymbol{a}), \tag{31c}$$

where $\mathfrak{A}(\boldsymbol{m})$ is the set of feasible allocation vectors defined in (21) and $\mathfrak{D}(\boldsymbol{m},\boldsymbol{a})$ is the set of feasible assignment and transmission vectors defined by (25)-(30). Note that $R^*_{\mathfrak{A},\mathfrak{D}}(\boldsymbol{m}) \geq R^*_{\mathfrak{A}}(\boldsymbol{m}) \geq R^*(\boldsymbol{m})$, where R^*, $R^*_{\mathfrak{A}}$ are defined in Section II.

Remark 1. *In the case of equal cache sizes, $R^*_{\mathfrak{A},\mathfrak{D}}(\boldsymbol{m})$ is equal to the worst-case delivery of the MaddahAli-Niesen caching scheme in [1], which was shown to be optimal for uncoded prefetching in [9]. Moreover, the solution obtained from the optimization problem is equivalent to the memory sharing solution proposed in [1].* ■

In order to evaluate the performance of the proposed caching scheme, we compare $R^*_{\mathfrak{A},\mathfrak{D}}(\boldsymbol{m})$ to the following lower bound on $R^*(\boldsymbol{m})$ which is derived in [4]. For given K, N, $\boldsymbol{M}$, and $M_1 \leq \cdots \leq M_K$, R^* is lower bounded by

$$\max_{s\in[K],\, l\in\left[\lceil\frac{N}{s}\rceil\right]} \frac{N-(N-Kl)^+}{l} - \frac{s\sum_{i=1}^{s+\gamma} M_i + \gamma(N-ls)^+}{l(s+\gamma)}, \tag{32}$$

where $\gamma := \min\left\{\left(\lceil\frac{N}{l}\rceil - s\right)^+, K-s\right\}$, and $(x)^+ := \max\{0,x\}$. This bound is tighter than the cut-set bound [3], given by

$$\max_{s\in[\min\{K,N\}]} \left\{ s - \frac{N\sum_{i=1}^{s} m_i}{\lfloor N/s\rfloor} \right\}. \tag{33}$$

B. Special Case: $\sum_{i=1}^K m_i \leq 1$

For the case where the sum of the cache sizes is less than or equal to the library size, i.e., $\sum_{i=1}^K m_i \leq 1$, we characterize explicitly the optimal solution of (31) as follows.

Proposition 1. *For $\sum_{i=1}^K m_i \leq 1$ and $m_1 \leq \cdots \leq m_K$, the minimum worst-case delivery load is given by*

$$R^*_{\mathfrak{A},\mathfrak{D}}(\boldsymbol{m}) = K - \sum_{j=1}^{K}(K-j+1)m_j,$$

where $a^*_{\{j\}} = m_j$, $v^*_{\{j\}} = 1-\sum_{i=1}^{j-1} m_i - (K-j+1)m_j$, and $v^*_{\{i,j\}} = u^{*\{i,j\}}_{\{i\}} = u^{*\{i,j\}}_{\{j\}} = \min\{a^*_{\{i\}}, a^*_{\{j\}}\}$. ■

Proof. (Outline) The sum of the constraints in (26) gives

$$\begin{aligned} R_{\mathfrak{A},\mathfrak{D}}(\boldsymbol{m}) &\geq K - \sum_{i=1}^{K} m_i - \sum_{\mathcal{T}\in 2^{[K]}-\{\}:|\mathcal{T}|\geq 2} (|\mathcal{T}|-1)\, v_{\mathcal{T}}, \\ &\geq K - \sum_{j=1}^{K}(K-j+1)m_j, \end{aligned}$$

since $\sum_{\mathcal{T}\in 2^{[K]}-\{\}:|\mathcal{T}|\geq 2} (|\mathcal{T}|-1)\, v_{\mathcal{T}} \leq \sum_{j=1}^{K}(K-j)\, m_j$, for $m_1 \leq \cdots \leq m_K$. Moreover, for $\sum_{i=1}^K m_i \leq 1$, $a^*_{\mathcal{S}}$, $v^*_{\mathcal{T}}$, and $u^{*\mathcal{T}}_{\mathcal{S}}$, represent a feasible solution that achieves the lower bound, i.e., $R^*_{\mathfrak{A},\mathfrak{D}}(\boldsymbol{m}) = K - \sum_{j=1}^{K}(K-j+1)m_j$. □

Using the genie-aided approach in [9], we can show that $R^*_{\mathfrak{A}}(\boldsymbol{m}) = R^*_{\mathfrak{A},\mathfrak{D}}(\boldsymbol{m})$, which means that the caching scheme in Proposition 1 is optimal under uncoded prefetching.

Proposition 2. *For $\sum_{i=1}^K m_i \leq 1$ and $m_1 \leq \cdots \leq m_K$,*

$$R^*_{\mathfrak{A}}(\boldsymbol{m}) \geq K - \sum_{j=1}^{K}(K-j+1)m_j, \tag{34}$$

which implies $R^*_{\mathfrak{A}}(\boldsymbol{m}) = R^*_{\mathfrak{A},\mathfrak{D}}(\boldsymbol{m})$. ■

Proof. (Outline) From the genie-aided approach in [9], we get

$$\begin{aligned} R^*_{\mathfrak{A}}(\boldsymbol{m}) &\geq \min_{\boldsymbol{a}\in\mathfrak{A}(\boldsymbol{m})} \sum_{j=1}^{K} \sum_{\mathcal{S}\in 2^{[K]}:\, \{1,\ldots,j\}\cap\mathcal{S}=\{\}} \boldsymbol{a}_{\mathcal{S}}, \\ &= \min_{\boldsymbol{a}\in\mathfrak{A}(\boldsymbol{m})} Ka_{\{\}} + \sum_{j=1}^{K-1} j \sum_{\mathcal{S}\in 2^{[K]}:\, \{j+1\}\in\mathcal{S},\{1,\ldots,j\}\cap\mathcal{S}=\{\}} \boldsymbol{a}_{\mathcal{S}}, \end{aligned}$$

and (34) is obtained by solving the dual linear program. □

C. Discussion

In our proposed scheme, we have formed $X_{\mathcal{T},\boldsymbol{d}}$ by using file pieces of equal size. Next, we remark that this restriction does not lead to a performance degradation.

Remark 2. *A delivery scheme with $\tilde{X}_{\mathcal{T},\boldsymbol{d}} = \oplus_{j\in\mathcal{T}} \varphi_{v_{\mathcal{T}}}\left(W^{\mathcal{T}}_{d_j}\right)$, where $\varphi_{v_{\mathcal{T}}}(.)$ pads zeros to subfiles with size less than $v_{\mathcal{T}}F$ bits, is equivalent to a delivery scheme in $\mathfrak{D}$ and both yield the same delivery load. For example, a multicast signal $\tilde{X}_{\{1,2\},\boldsymbol{d}} = \varphi_{v_{\{1,2\}}}\left(W^{\{1,2\}}_{d_1,\{2\}}\right) \oplus \varphi_{v_{\{1,2\}}}\left(W^{\{1,2\}}_{d_2,\{1\}}\right)$, where $a_{\{2\}} > a_{\{1\}}$ and $\varphi_{v_{\{1,2\}}}(.)$ appends $a_{\{2\}} - a_{\{1\}}$ zeros to $W^{\{1,2\}}_{d_2,\{1\}}$, is equivalent to a multicast signal $X_{\{1,2\},\boldsymbol{d}}$ and a unicast signal $X_{\{2\},\boldsymbol{d}}$, with sizes $a_{\{1\}}F$ bits, and $\left(a_{\{2\}} - a_{\{1\}}\right)F$ bits, respectively.* ■

Utilizing Remark 2 enables us to provide the following proposition comparing the performance of our propose scheme to that of the decentralized scheme in [3].

Proposition 3. *For given $N, K, \boldsymbol{m}$ and $m_1 \leq \cdots \leq m_K$,*

$$R^*_{\mathfrak{A},\mathfrak{D}}(\boldsymbol{m}) \leq R_{dec}(\boldsymbol{m}) = \sum_{i=1}^{K}\prod_{j=1}^{i}(1-m_j). \tag{35}$$

■

Proof. The decentralized placement scheme [3] is represented by $a_{\mathcal{S}} = \prod_{i\in\mathcal{S}} m_i \prod_{i\in\mathcal{S}^c}(1-m_i)$, which belongs to $\mathfrak{A}$. The decentralized delivery scheme [3] is equivalent to a delivery scheme in $\mathfrak{D}$ by Remark 2. Thus, the decentralized scheme in [3] is a feasible (but not necessarily optimal) solution to the optimization problem in (31). □

VI. Numerical results

In this section, we first present a numerical example that explains how to interpret the solution of the optimization

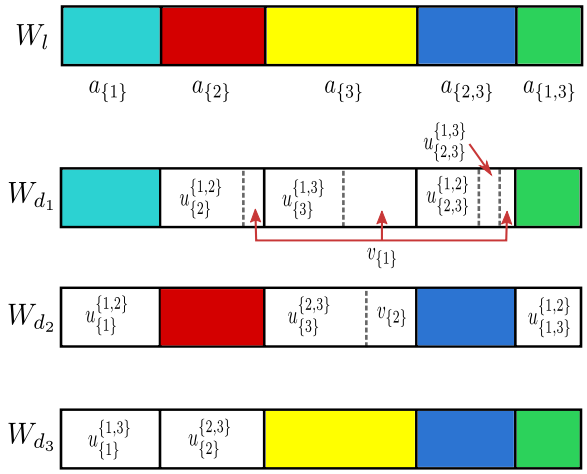

Fig. 4: Illustration of the optimal solution of (31) for a three-user system with cache sizes $m = [0.3,\ 0.4,\ 0.6]$.

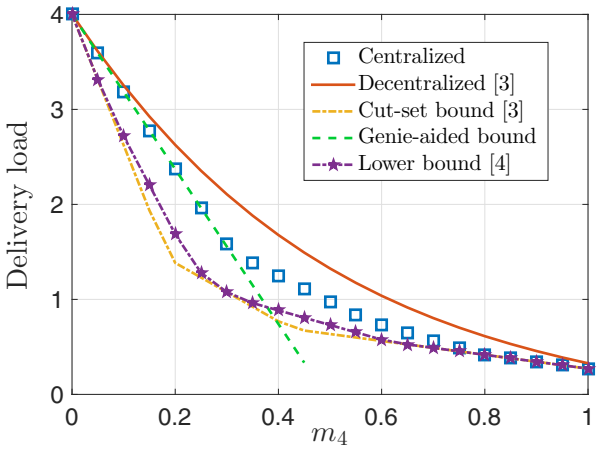

Fig. 5: Comparing $R^*_{\mathfrak{A},\mathfrak{D}}(\boldsymbol{m})$, $R_{\text{dec}}(\boldsymbol{m})$, cut-set and genie-aided lower bounds, for $K = 4$, and $m_k = 0.9\, m_{k+1}$.

problem in (31). Then, we compare numerically the centralized delivery load obtained from (31) with the decentralized delivery load and the existing lower bounds.

Example 1. *Consider a caching system with $K = N = 3$, and $\boldsymbol{M} = [0.9,\ 1.2,\ 1.8]$, i.e., $\boldsymbol{m} = [0.3,\ 0.4,\ 0.6]$. The caching scheme obtained from (31), is characterized as follows.*
Placement phase: *Every file $W^{(l)}$ is divided into five subfiles, such that $a_{\{1\}} = 0.1958$, $a_{\{2\}} = 0.2042$, $a_{\{3\}} = 0.3$, $a_{\{2,3\}} = 0.1958$, and $a_{\{1,3\}} = 0.1042$, as shown in Fig. 4.*
Delivery phase: *We have the following transmissions*

1) *$X_{\{1,2\},\boldsymbol{d}}$ delivers the pieces corresponding to $u_{\{2\}}^{\{1,2\}} = 0.17457$, $u_{\{2,3\}}^{\{1,2\}} = 0.12543$ to user 1, and $u_{\{1\}}^{\{1,2\}} = 0.1958$, $u_{\{1,3\}}^{\{1,2\}} = 0.1042$ to user 2, i.e., $v_{\{1,2\}} = 0.3$.*
2) *$X_{\{1,3\},\boldsymbol{d}}$ delivers the pieces corresponding to $u_{\{3\}}^{\{1,3\}} = 0.15265$, $u_{\{2,3\}}^{\{1,3\}} = 0.04315$ to user 1, and $u_{\{1\}}^{\{1,3\}} = 0.1958$ to user 3, i.e., $v_{\{1,3\}} = 0.1958$.*
3) *$X_{\{2,3\},\boldsymbol{d}}$ delivers the piece corresponding to $u_{\{3\}}^{\{2,3\}} = 0.2024$ to user 2, and the piece corresponding to $u_{\{2\}}^{\{2,3\}} = 0.2024$ to user 3, i.e., $v_{\{2,3\}} = 0.2024$.*
4) *The unicast transmissions to users 1 and 2 are represented by $v_{\{1\}} = 0.2024$ and $v_{\{2\}} = 0.0958$, respectively.*

At the end of transmission, user k is able to decode file W_{d_k}, and the minimum worst-case delivery load $R^*_{\mathfrak{A},\mathfrak{D}}(\boldsymbol{m}) = 1$. The breakdown of the requested files is illustrated in Fig. 4. ■

In Fig. 5, we compare the centralized delivery load $R^*_{\mathfrak{A},\mathfrak{D}}(\boldsymbol{m})$, with the decentralized delivery load $R_{\text{dec}}(\boldsymbol{m})$, the lower bound in (32), the cut-set bound in (33), and the genie-aided bound for uncoded prefetching in (34), for $K = 4$, and $m_k = 0.9^{4-k}\, m_4$. We observe that $R^*_{\mathfrak{A},\mathfrak{D}}(\boldsymbol{m})$ coincides with the genie-aided lower bound for $\sum_{k=1}^{4} m_k \leq 1$. Moreover, $R^*_{\mathfrak{A},\mathfrak{D}}(\boldsymbol{m})$ coincides with the cut-set bound for large memories, and $R^*_{\mathfrak{A},\mathfrak{D}}(\boldsymbol{m}) \leq R_{\text{dec}}(\boldsymbol{m})$.

VII. Conclusions

In this paper, we have considered a centralized coded caching system with heterogeneous cache sizes. We have proposed a caching scheme taking into account the unequal cache sizes. We have put forward an optimization framework for minimizing the worst-case delivery load by optimizing the parameters of the caching scheme, for given cache memory sizes. We have shown that our caching scheme is optimal under uncoded prefetching for $\sum_{k=1}^{K} m_k \leq 1$. We have illustrated the caching scheme by a numerical example and compared our centralized delivery load with the decentralized delivery load derived in [3], and the lower bounds in [3], [4].

This work is a step towards developing caching schemes for systems with practical considerations, such as wireless systems where the users have distinct download rates and non-uniform resources. Future directions include optimizing over cache sizes and consideration of incomplete/inaccurate demand profiles.

References

[1] M. A. Maddah-Ali and U. Niesen, "Fundamental limits of caching," *IEEE Trans. Info. Theory*, vol. 60, no. 5, pp. 2856–2867, Mar. 2014.
[2] ——, "Decentralized coded caching attains order-optimal memory-rate tradeoff," *IEEE/ACM Trans. Netw.*, vol. 23, no. 4, pp. 1029–1040, Aug. 2015.
[3] S. Wang, W. Li, X. Tian, and H. Liu, "Coded caching with heterogenous cache sizes," *arXiv:1504.01123*, 2015.
[4] M. M. Amiri, Q. Yang, and D. Gündüz, "Decentralized coded caching with distinct cache capacities," *arXiv:1611.01579*, 2016.
[5] N. Karamchandani, U. Niesen, M. A. Maddah-Ali, and S. N. Diggavi, "Hierarchical coded caching," *IEEE Trans. Info. Theory*, vol. 62, no. 6, pp. 3212–3229, Apr. 2016.
[6] M. Ji, G. Caire, and A. F. Molisch, "Fundamental limits of caching in wireless D2D networks," *IEEE Trans. Info. Theory*, vol. 62, no. 2, pp. 849–869, Feb. 2016.
[7] F. Xu, M. Tao, and K. Liu, "Fundamental tradeoff between storage and latency in cache-aided wireless interference networks," *arXiv:1605.00203*, 2016.
[8] N. Naderializadeh, M. A. Maddah-Ali, and A. S. Avestimehr, "Fundamental limits of cache-aided interference management," *arXiv:1602.04207*, 2016.
[9] Q. Yu, M. A. Maddah-Ali, and A. S. Avestimehr, "The exact rate-memory tradeoff for caching with uncoded prefetching," *arXiv:1609.07817*, 2016.