Optimization of Heterogeneous Caching Systems with Rate Limited Links

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*—This paper considers centralized coded caching, where the server not only designs the users' cache contents, but also assigns their cache sizes under a total cache memory budget. The server is connected to each user via a link of given finite capacity. For given link capacities and total memory budget, we minimize the worst-case delivery completion time by jointly optimizing the cache sizes, the cache placement and delivery schemes. The optimal memory allocation and caching scheme are characterized explicitly for the case where the total memory budget is smaller than that of the server library. Numerical results confirm the savings in delivery time obtained by optimizing the memory allocation.**

I. Introduction

Caching [1]–[4] exploits under utilization of network resources during off-peak hours in order to reduce network traffic during congestion periods. In caching, the *placement phase* refers to low traffic periods during which contents are prefetched and stored at the end users' cache memories, and the *delivery phase* refers to congestion periods during which we try to reduce the traffic needed for the delivery of the requested files. Reference [1] has shown that the delivery load can be significantly reduced by using *coded caching*, in which the cache contents are designed in order to create multicast opportunities during the delivery phase.

Whereas references [1]–[4] have considered a noiseless setup for delivery and followed a source-channel separation approach, recent work in design of coded caching schemes takes into account the channel noise. In particular, references [5]–[7] consider degraded broadcast channels with cache-aided receivers and propose a joint cache-channel coding approach showing improvement in fundamental limits. A middle ground between [1]–[4] and [5]–[7] is to consider capacity limited links in lieu of a particular channel description, a modeling approach we shall follow in this paper.

Recently, in reference [8], we have proposed cache placement and delivery schemes for centralized caching with unequal fixed cache sizes, where the network links have equal finite capacities. We have characterized the optimal caching scheme that minimizes the worst-case delivery load for given cache sizes. Different from [8], in this paper, we consider a caching system, where the server is connected to the users via a multicast network that consists of rate limited links of *different* capacities similar to the model in [9]. In particular,

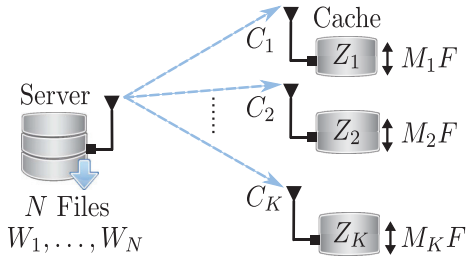

Fig. 1: Heterogeneous centralized caching system.

the link from the server to user k has a fixed capacity C_k bits per channel use, see Fig. 1. In turn, a multicast signal to the users in the set $\mathcal{T}$ needs to be transmitted with a rate less than $\min_{j\in\mathcal{T}} C_j$ [9]. Furthermore, the server controls the users' cache sizes subject to a cache memory budget, which can be implemented via communicating the optimal cache size to each user via a control channel. In other words, the server, e.g., a base station, determines the optimal fractions of the users' physical memories to be dedicated to caching in order to minimize the worst-case delivery completion time (DCT).

More specifically, building on the caching scheme in [8], we formulate the problem of minimizing *the worst-case delivery completion time* by jointly optimizing the caching scheme and the users' cache sizes subject to a cache memory budget constraint. The optimal memory allocation and caching scheme are obtained by solving a linear program. In addition, for the case where the cache memory budget is less than or equal to the library size at the server, we find closed form expressions for the optimal cache sizes, placement and delivery policies. The optimal solution balances between allocating larger cache memories to users with low capacities and equalizing the cache memory sizes. The former implies transmitting fewer number of bits in order to satisfy their demands, while the latter maximizes the multicast gain. Additionally, we compare the optimal memory allocation and caching scheme with the state-of-art, showing the improvement in delivery time.

Notation: Vectors are represented by boldface letters, $\oplus$ refers to bitwise XOR operation, $|W|$ denotes the size of W, $[K] := \{1, \ldots, K\}$, and $2^{[K]}$ denotes the power set of $[K]$.

978-1-4673-8999-0/17/$31.00 ©2017 IEEE

II. System Model

We consider a multicast network connecting one server to K users via links of different capacities, see Fig. 1. The link between the server and user k has capacity C_k bits per channel use, which we refer to as the *download rate* at user k. The download rates vector is denoted by $\boldsymbol{C} = [C_1, \ldots, C_K]$. Additionally, the server contains a library of N files, $W_1, \ldots, W_N$, each with size F bits. The size of the users' cache memories are determined by the server. In particular, the server allocates $M_k F$ bits to user k such that $\sum_{k=1}^{K} M_k F \leq m_{\text{tot}} NF$ bits, where m_{tot} is the cache memory budget normalized by the library size NF. We also define $m_k = M_k/N$, to denote the memory size at user k normalized by the library size NF. We assume that the number of files is greater than or equal the number of users, i.e., $N \geq K$, and $M_k \in [0, N],\ \forall k \in [K]$, which implies $m_k \in [0, 1],\ \forall k \in [K]$. We denote the memory size vector by $\boldsymbol{M} = [M_1, \ldots, M_K]$ and its normalized version by the library size by $\boldsymbol{m} = [m_1, \ldots, m_K]$.

The system operates over two phases: placement phase where the server populates the users' cache memories, and delivery phase where the server delivers the files requested by the users. The users' demands are unknown until the beginning of the delivery phase. In the placement phase, 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 and the users' demands are assumed to be 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 [1]. In order to deliver the requested files, the server transmits a sequence of unicast/multicast signals, $X_{\mathcal{T},\boldsymbol{d}}$, where $\mathcal{T} \subset [K]$. User k should be able to reconstruct W_{d_k} from the signals $X_{\mathcal{T},\boldsymbol{d}}$, $k \in \mathcal{T}$, $\mathcal{T} \subset [K]$ and Z_k.

In this work, we consider the set of caching policies $\mathfrak{A}$ that satisfies the following assumptions:

1) We consider *uncoded prefetching* [10], where the server places uncoded data at the users' cache memories, i.e., there is no coding over files.
2) Under uniform demands, the cache memory at user k is divided equally over the files, i.e., $m_k F$ bits are dedicated to each file.

A cache placement policy in $\mathfrak{A}$ is identified by an allocation vector $\boldsymbol{a}$ which represents the partitions of the files stored exclusively at each subset of users $\mathcal{S} \subset [K]$.

On the other hand, the set of delivery schemes $\mathfrak{D}$ satisfies the following assumptions:

1) A multicast signal $X_{\mathcal{T},\boldsymbol{d}}$ is created by XORing $|\mathcal{T}|$ file pieces of equal size, and $|X_{\mathcal{T},\boldsymbol{d}}| = v_{\mathcal{T}} F$ bits.
2) A unicast signal $X_{\{k\},\boldsymbol{d}}$ delivers the missing pieces to user k, i.e., the pieces that are not delivered by the multicast signals and are not stored at user k.
3) $X_{\mathcal{T},\boldsymbol{d}}$ is intended for the users in $\mathcal{T}$, hence it is transmitted with a rate less than or equal to $\min_{j\in\mathcal{T}} C_j$ [9].

It worth noting that the third assumption implies that the users outside the set $\mathcal{T}$ may not be able to decode the signal $X_{\mathcal{T},\boldsymbol{d}}$, as their decoding rates may be lower than $\min_{j\in\mathcal{T}} C_j$. A delivery policy in $\mathfrak{D}$ is identified by a transmission vector $\boldsymbol{v}$, representing the size of the transmitted signals, and an assignment vector $\boldsymbol{u}$ representing the structure of the transmitted signals.

Our goal in this work is to minimize the worst-case delivery completion time (DCT) over all possible demand instances, which is defined as follows.

Definition 1. *The worst-case delivery completion time (DCT) under a cache placement policy in $\mathfrak{A}$ and a delivery policy in $\mathfrak{D}$, is defined as* $\Theta_{\mathfrak{A},\mathfrak{D}} := \max_{\boldsymbol{d}} \Theta_{\boldsymbol{d},\mathfrak{A},\mathfrak{D}} = \sum_{\mathcal{T}\in 2^{[K]}-\{\}} \frac{v_{\mathcal{T}}}{\min_{j\in\mathcal{T}} C_j}$. ■

III. Problem Formulation

For given rates vector $\boldsymbol{C}$, and normalized cache budget m_{tot}, we minimize the worst-case delivery completion time (DCT), by jointly optimizing the caching scheme and the users' cache sizes. In particular, the following optimization problem identifies the minimum worst-case DCT, $\Theta^*_{\mathfrak{A},\mathfrak{D}}(m_{\text{tot}}, \boldsymbol{C})$, the optimal memory allocation, and the optimal caching scheme in $\mathfrak{A}, \mathfrak{D}$, i.e., the optimal values for $\boldsymbol{m}$, $\boldsymbol{a}$, $\boldsymbol{v}$, and $\boldsymbol{u}$.

$$\textbf{\textit{O1}}: \min_{\boldsymbol{a},\boldsymbol{u},\boldsymbol{v},\boldsymbol{m}} \quad \sum_{\mathcal{T}\in 2^{[K]}-\{\}} \frac{v_{\mathcal{T}}}{\min_{j\in\mathcal{T}} C_j} \tag{1a}$$

$$\text{subject to} \quad \boldsymbol{a} \in \mathfrak{A}(\boldsymbol{m}), \tag{1b}$$

$$(\boldsymbol{v}, \boldsymbol{u}) \in \mathfrak{D}(\boldsymbol{m}, \boldsymbol{a}), \tag{1c}$$

$$\sum_{k=1}^{K} m_k \leq m_{\text{tot}}, \tag{1d}$$

$$0 \leq m_k \leq 1,\ \forall\, k \in [K], \tag{1e}$$

where $\mathfrak{A}(\boldsymbol{m})$ is the set of feasible allocation vectors defined in (5) and $\mathfrak{D}(\boldsymbol{m}, \boldsymbol{a})$ is the set of feasible assignment and transmission vectors defined by (8)-(13). The caching scheme is detailed in [8], and recapped briefly in Section V.

IV. Cache Size Optimization

In general, the optimal memory allocation and caching scheme are obtained by solving the linear program in (1), which are illustrated numerically in Section VI. In addition, for the case where $m_{\text{tot}} \leq 1$, a closed form solution is possible to be obtained which we present next.

A. Optimal Solution for $m_{tot} \leq 1$

For the case, where $m_{\text{tot}} \leq 1$, the optimal memory allocation balances between uniform memory allocation and allocating larger cache memories to users with low decoding rates. In particular, the cache memory budget m_{tot} is allocated uniformly over users $\{1, \ldots, q\}$, where q is determined by $\boldsymbol{C}$ as illustrated in the following theorem.

Theorem 1. *For $C_1 \leq \cdots \leq C_K$ and $m_{tot} \leq 1$, the minimum worst-case delivery completion time*

$$\Theta^*_{\mathfrak{A},\mathfrak{D}}(m_{tot}, \boldsymbol{C}) = \sum_{j=1}^{K} \frac{1}{C_j} - \max_{i\in[K]} \left\{ \sum_{j=1}^{i} \frac{j\, m_{tot}}{i\, C_j} \right\},$$

and the optimal memory allocation is $m_1^* = \cdots = m_q^* = \frac{m_{tot}}{q}$, *where* $q = \text{argmax}_{i\in[K]} \left\{ \sum_{j=1}^{i} \frac{j\, m_{tot}}{i\, C_j} \right\}$. *Moreover, if the solution is not unique, i.e.,* $q \in \{q_1, \ldots, q_L\}$, *for some* $L \leq K$, *then* $\boldsymbol{m}^* = \sum_{i=1}^{L} \alpha_i \left[\frac{m_{tot}}{q_i}, \ldots, \frac{m_{tot}}{q_i}, 0, \ldots, 0 \right]$, *where* $\sum_{i=1}^{L} \alpha_i = 1$ *and* $\alpha_i \geq 0$. ■

The proof of Theorem 1 is provided in Appendix A. In the following, we describe the optimal caching scheme that achieves the DCT in Theorem 1. In particular, the optimal cache placement scheme is to split each file W_l into $K+1$ subfiles, $\tilde{W}_{l,\{\}}, \tilde{W}_{l,\{1\}}, \ldots, \tilde{W}_{l,\{K\}}$, such that $|\tilde{W}_{l,\{\}}| = (1 - m_{\text{tot}})F$, $|\tilde{W}_{l,\{j\}}| = m_j^* F$ and user j caches subfiles $\tilde{W}_{l,\{j\}},\ \forall\, l \in [N]$. On the other hand, the optimal delivery scheme for this case, considers only the pairwise multicast signals $X_{\{i,j\},\boldsymbol{d}}$ and the unicast signals $X_{\{j\},\boldsymbol{d}}$. Specifically, a multicast signal to users $\{i, j\}$ is defined by

$$X_{\{i,j\},\boldsymbol{d}} = W_{d_i}^{\{i,j\}} \oplus\ W_{d_j}^{\{i,j\}}, \tag{2}$$

where $W_{d_i}^{\{i,j\}} \subset \tilde{W}_{d_i,\{j\}}$, $W_{d_j}^{\{i,j\}} \subset \tilde{W}_{d_j,\{i\}}$, and $|W_{d_i}^{\{i,j\}}| = |W_{d_j}^{\{i,j\}}| = \min\{m_i^*, m_j^*\}F$. Additionally, a unicast signal to user j completes the missing pieces from W_{d_j}, which is defined by

$$X_{\{j\},\boldsymbol{d}} = W_{d_j} - \tilde{W}_{d_j,\{j\}} - \bigcup_{i=1, i\neq j}^{K} W_{d_j}^{\{i,j\}}, \tag{3}$$

i.e., $|X_{\{j\},\boldsymbol{d}}| = (1 - m_j^* - \sum_{i=1, i\neq j}^{K} \min\{m_i^*, m_j^*\})F$.

B. Uniform Memory Allocation

In this section, we consider uniform cache allocation which maximizes the multicast opportunities, without taking into account the impact of different channels on the DCT, and conclude its suboptimality. Lemma 1 considers uniform memory allocation combined with the MaddahAli-Niesen caching scheme [1].

Lemma 1. *For* $m_{tot} \in [K]$ *and uniform memory allocation, the MaddahAli-Niesen caching scheme [1] is a feasible solution to (1). Moreover, for* $C_1 \leq C_2 \leq \cdots \leq C_K$, *the worst-case delivery completion time under this scheme is given by*

$$\Theta_{unif}(m_{tot}, \boldsymbol{C}) = \frac{1}{\binom{K}{m_{tot}}} \sum_{j=1}^{K-m_{tot}} \frac{\binom{K-j}{m_{tot}}}{C_j}. \tag{4}$$

The proof is discussed in Appendix B. From Lemma 1, we observe that (4) yields an upper bound on the minimum worst-case DCT.

Corollary 1. *For given* $\boldsymbol{C}$ *and* m_{tot}, $\Theta^*_{\mathfrak{A},\mathfrak{D}}(m_{tot}, \boldsymbol{C})$ *is less than or equal to* $\Theta_{unif}(m_{tot}, \boldsymbol{C})$, *which is obtained by uniform memory allocation, i.e.,* $m_j = m_{tot}/K, \forall j \in [K]$, *and applying the MaddahAli-Niesen caching scheme [1].* ■

V. Caching Model

We adopt the cache placement scheme proposed in [8], which we compactly summarize here for completeness. The caching scheme defines the sets $\mathfrak{A}(\boldsymbol{m})$ and $\mathfrak{D}(\boldsymbol{m}, \boldsymbol{a})$.

A. Cache Placement Phase

For a system with K users, each file W_l is partitioned into 2^K subfiles, which are denoted by $\tilde{W}_{l,\mathcal{S}}, \mathcal{S} \in 2^{[K]}$, and the fraction of W_l stored at the users in $\mathcal{S}$ is represented by the allocation variable $a_{\mathcal{S}} \in [0, 1]$. Additionally, the aforementioned partitioning is symmetric over all files, i.e., $|\tilde{W}_{l,\mathcal{S}}| = a_{\mathcal{S}} F$ bits, $\forall l \in [N]$. Therefore, for a given $\boldsymbol{m}$, the set of feasible placement schemes $\mathfrak{A}(\boldsymbol{m})$ is defined as follows

$$\left\{ \boldsymbol{a} \in [0,1]^{2^K} \middle| \sum_{\mathcal{S}\in 2^{[K]}} a_{\mathcal{S}} = 1, \sum_{\mathcal{S}\in 2^{[K]}\,:\, k\in\mathcal{S}} a_{\mathcal{S}} \leq m_k, \forall k \in [K] \right\}, \tag{5}$$

and the content of the cache memory at user k is given by

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

B. Delivery Phase

The delivery scheme is defined by the unicast/multicast signals $X_{\mathcal{T},\boldsymbol{d}}, \mathcal{T} \in 2^{[K]} - \{\}$, where $2^{[K]} - \{\}$ denotes all possible transmission sets. In particular, user k is able to reconstruct W_{d_k} from the signals $X_{\mathcal{T},\boldsymbol{d}}$, $k \in \mathcal{T}$, $\mathcal{T} \in 2^{[K]} - \{\}$ and its cache content Z_k. The signal intended to the users in $\mathcal{T}$ is formed by XORing pieces from $W_{d_j}, j \in \mathcal{T}$, each with size $v_{\mathcal{T}} F$ bits, i.e.,

$$X_{\mathcal{T},\boldsymbol{d}} = \oplus_{j\in\mathcal{T}}\ W_{d_j}^{\mathcal{T}}, \tag{6}$$

where $W_{d_j}^{\mathcal{T}} \subset W_{d_j}$ is the piece of the file requested by j and delivered via $X_{\mathcal{T},\boldsymbol{d}}$. The unicast signal to j delivers the pieces of W_{d_j} that had not been delivered by multicast signals and are not available locally.

In order to guarantee that each user $j \in \mathcal{T}$ is able to extract its requested piece $W_{d_j}^{\mathcal{T}}$ from the signal $X_{\mathcal{T},\boldsymbol{d}}$, the remaining pieces that form $X_{\mathcal{T},\boldsymbol{d}}$ must be contained in $Z_j,\ j \in \mathcal{T}$. Define $W_{d_j,\mathcal{S}}^{\mathcal{T}}$ as the subset of $W_{d_j}^{\mathcal{T}}$ stored at the users in $\mathcal{S}$, i.e.,

$$W_{d_j}^{\mathcal{T}} := \bigcup_{\mathcal{S}\in\mathcal{B}_j^{\mathcal{T}}} W_{d_j,\mathcal{S}}^{\mathcal{T}}, \tag{7}$$

where $\mathcal{B}_j^{\mathcal{T}} := \left\{\mathcal{S} \in 2^{[K]} : \mathcal{T} - \{j\} \subset \mathcal{S}, j \notin \mathcal{S}\right\}$, for $j \in \mathcal{T}$, are the sets storing the side information at $\mathcal{T} - \{j\}$ and not available at j. Additionally, denote all allocation sets related to $\mathcal{T}$ by $\mathcal{B}^{\mathcal{T}} := \bigcup_{j\in\mathcal{T}} \mathcal{B}_j^{\mathcal{T}}$. Moreover, $|W_{d_j,\mathcal{S}}^{\mathcal{T}}| = u_{\mathcal{S}}^{\mathcal{T}} F$ bits, which implies that the fraction of $\tilde{W}_{d_j,\mathcal{S}}$ involved in the multicast transmission to the users in $\mathcal{T}$, is represented by the assignment variable $u_{\mathcal{S}}^{\mathcal{T}} \in [0, a_{\mathcal{S}}]$. Hence, the structure of $W_{d_j}^{\mathcal{T}}$ is represented by

$$\sum_{\mathcal{S}\in\mathcal{B}_j^{\mathcal{T}}} u_{\mathcal{S}}^{\mathcal{T}} = v_{\mathcal{T}},\ \forall \mathcal{T} \in 2^{[K]} - \{\},\ \forall\, j \in \mathcal{T}. \tag{8}$$

The amount of side information stored at the users limits the size of the multicast signals $X_{\mathcal{T},\boldsymbol{d}}$. Specifically, the amount of side information stored at the users in $\mathcal{S}'$ and not available at user j, imposes the following constraints

$$\sum_{\mathcal{T}\in 2^{[K]}-\{\}\,:\,\{j\}\cup\mathcal{S}'\subset\mathcal{T}} v_{\mathcal{T}} \leq \sum_{\mathcal{S}\in 2^{[K]}:\ \mathcal{S}'\subset\mathcal{S}, j\notin\mathcal{S}} a_{\mathcal{S}},\ \forall j \notin \mathcal{S}',$$

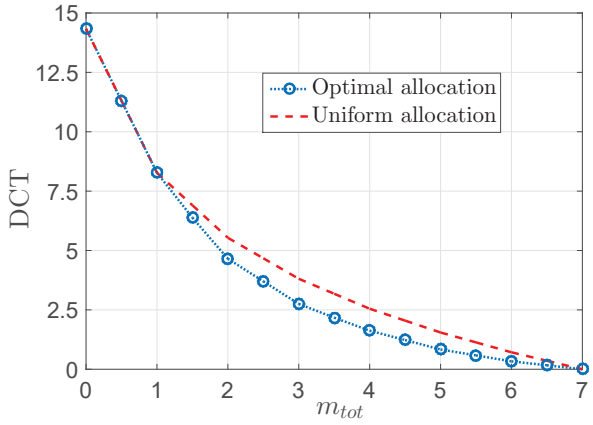

Fig. 2: Comparing $\Theta^*_{\mathfrak{A},\mathfrak{D}}(m_{\text{tot}}, \boldsymbol{C})$ and $\Theta_{\text{unif}}(m_{\text{tot}}, \boldsymbol{C})$ for $K = 7$, and $\boldsymbol{C} = [0.2,\ 0.4,\ 0.6,\ 0.6,\ 0.8,\ 0.8,\ 1]$.

$$\forall\, \mathcal{S}' \in \left\{\tilde{\mathcal{S}} \in 2^{[K]} : 1 \leq |\tilde{\mathcal{S}}| \leq K-2\right\}. \tag{9}$$

Moreover, in order to prevent the transmission of redundant bits from $\tilde{W}_{d_j,\mathcal{S}}$ to user j, we need to ensure

$$\sum_{\mathcal{T}\in 2^{[K]}-\{\} \,:\, j\in\mathcal{T}, \mathcal{T}\cap\mathcal{S}\neq\{\}} u_{\mathcal{S}}^{\mathcal{T}} \leq a_{\mathcal{S}},\ \forall\, j \notin \mathcal{S},$$
$$\forall\, \mathcal{S} \in \left\{\tilde{\mathcal{S}} \in 2^{[K]} :\ 2 \leq |\tilde{\mathcal{S}}| \leq K-1\right\}. \tag{10}$$

Finally, the users' demands are satisfied by

$$\sum_{\mathcal{T}\in 2^{[K]}-\{\} \,:\, k\in\mathcal{T}} v_{\mathcal{T}} \geq 1 - m_k,\ \forall\, k \in [K], \tag{11}$$

since m_k represents the local caching gain.

In summary, a delivery scheme is defined by the assignment and transmission variables, identified by the assignment vector $\boldsymbol{u}$ and transmission vector $\boldsymbol{v}$, respectively. The set of feasible delivery schemes for given $\boldsymbol{m}$ and $\boldsymbol{a}$, which we denote by $\mathfrak{D}(\boldsymbol{m}, \boldsymbol{a})$, must satisfy (8)-(11), and

$$0 \leq u_{\mathcal{S}}^{\mathcal{T}} \leq a_{\mathcal{S}},\ \forall\, \mathcal{T} \in 2^{[K]} - \{\},\ \forall\, \mathcal{S} \in \mathcal{B}^{\mathcal{T}}, \tag{12}$$

$$0 \leq v_{\mathcal{T}} \leq 1,\ \forall\, \mathcal{T} \in 2^{[K]} - \{\}. \tag{13}$$

Remark 1. *For $\sum_{k=1}^{K} m_k \leq 1$, the general caching scheme reduces to the simple scheme described in Section IV-A.* ■

VI. Numerical Results

In this section, we present our numerical results solving (1), and compare the optimal solution with the MaddahAli-Niesen caching scheme under uniform memory allocation.

Example 1. *Consider a three-user caching system with memory budget $m_{tot} = 1$, and $C_1 \leq C_2 \leq C_3$, which implies*

$$\Theta_{unif}(1, \boldsymbol{C}) = \frac{1}{\binom{3}{1}} \sum_{j=1}^{2} \frac{\binom{3-j}{1}}{C_j} = \frac{1}{3}\left(\frac{2}{C_1} + \frac{1}{C_2}\right),$$

and $q = \text{argmax}_{i\in[3]} \left\{\sum_{j=1}^{i} \frac{j}{i\, C_j}\right\}$. We consider the following cases for the link rates:

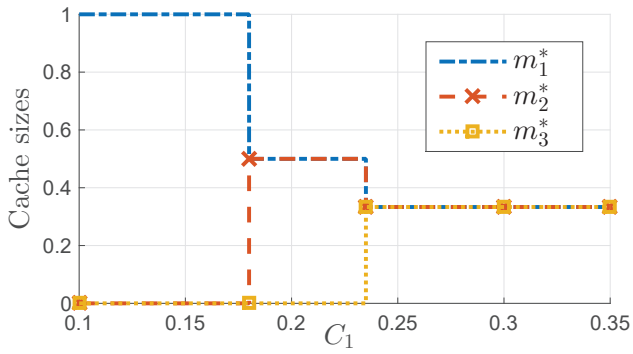

Fig. 3: The optimal memory allocations for $m_{\text{tot}} = 1, C_2 = 0.35$ and $C_3 = 0.6$.

1) *For $\boldsymbol{C} = [0.2,\ 0.4,\ 0.5]$, we get $q = 3$, hence the optimal solution is the MaddahAli-Niesen caching scheme under uniform memory allocation. In particular, we have $\Theta_{unif} = \Theta^*_{\mathfrak{A},\mathfrak{D}} = 4.1667$, $\boldsymbol{m}^* = [1/3,\ 1/3,\ 1/3]$, and the optimal caching scheme is given by*

$$a^*_{\{1\}} = a^*_{\{2\}} = a^*_{\{3\}} = 1/3,$$
$$v^*_{\{1,2\}} = u^{*\{1,2\}}_{\{1\}} = u^{*\{1,2\}}_{\{2\}} = 1/3,$$
$$v^*_{\{1,3\}} = u^{*\{1,3\}}_{\{1\}} = u^{*\{1,3\}}_{\{3\}} = 1/3,$$
$$v^*_{\{2,3\}} = u^{*\{2,3\}}_{\{2\}} = u^{*\{2,3\}}_{\{3\}} = 1/3.$$

2) *For $\boldsymbol{C} = [0.3,\ 0.3,\ 0.6]$, we get $q \in \{2, 3\}$, i.e., the optimal solution is not unique. In particular, we have $\boldsymbol{m}^* = [\frac{\alpha}{2} + \frac{1-\alpha}{3},\ \frac{\alpha}{2} + \frac{1-\alpha}{3},\ \frac{1-\alpha}{3}]$, where $\alpha \in [0, 1]$, and $\Theta^*_{\mathfrak{A},\mathfrak{D}} = \Theta_{unif} = 3.3333$, e.g., for $\alpha = 0.3082$, $\boldsymbol{m}^* = [0.3847,\ 0.3847,\ 0.2306]$ and the optimal caching scheme is given by*

$$a^*_{\{1\}} = a^*_{\{2\}} = 0.3847,\ a^*_{\{3\}} = 0.2306,$$
$$v^*_{\{1,2\}} = u^{*\{1,2\}}_{\{1\}} = u^{*\{1,2\}}_{\{2\}} = 0.3847,$$
$$v^*_{\{1,3\}} = u^{*\{1,3\}}_{\{1\}} = u^{*\{1,3\}}_{\{3\}} = 0.2306,$$
$$v^*_{\{2,3\}} = u^{*\{2,3\}}_{\{2\}} = u^{*\{2,3\}}_{\{3\}} = 0.2306,\ v^*_{\{3\}} = 0.3082.$$

3) *For $\boldsymbol{C} = [0.2,\ 0.3,\ 0.6]$, we get $q = 2$, hence $\boldsymbol{m}^* = [0.5,\ 0.5,\ 0]$ and the optimal caching scheme is given by $a^*_{\{1\}} = a^*_{\{2\}} = 0.5$, $v^*_{\{1,2\}} = u^{*\{1,2\}}_{\{1\}} = u^{*\{1,2\}}_{\{2\}} = 0.5$, $v^*_{\{3\}} = 1$, which results in $\Theta^*_{\mathfrak{A},\mathfrak{D}} = 4.1667$. On the other hand, $\Theta_{unif} = 4.4444$.* ■

In Fig. 2, we compare $\Theta^*_{\mathfrak{A},\mathfrak{D}}(m_{\text{tot}}, \boldsymbol{C})$ with $\Theta_{\text{unif}}(m_{\text{tot}}, \boldsymbol{C})$ for $K = 7$, and $\boldsymbol{C} = [0.2,\ 0.4,\ 0.6,\ 0.6,\ 0.8,\ 0.8,\ 1]$. We observe that $\Theta^*_{\mathfrak{A},\mathfrak{D}}(m_{\text{tot}}, \boldsymbol{C}) \leq \Theta_{\text{unif}}(m_{\text{tot}}, \boldsymbol{C})$, and for $m_{\text{tot}} \leq 1$, we have $\text{argmax}_{i\in[K]} \left\{\sum_{j=1}^{i} (j\, m_{\text{tot}})/(i C_j)\right\} = K$, which implies $\Theta^*_{\mathfrak{A},\mathfrak{D}}(m_{\text{tot}}, \boldsymbol{C}) = \Theta_{\text{unif}}(m_{\text{tot}}, \boldsymbol{C})$. Moreover, Fig. 3 shows the optimal memory allocation versus C_1 for $m_{\text{tot}} = 1, C_2 = 0.35$ and $C_3 = 0.6$, we observe that the memory allocated to user 1 is non-increasing in its link capacity.

In Fig. 4 and 5, we show the optimal memory allocation for $\boldsymbol{C} = [0.2,\ 0.2,\ 0.2,\ 0.5,\ 0.6,\ 0.7,\ 0.7]$ and $\boldsymbol{C} =$

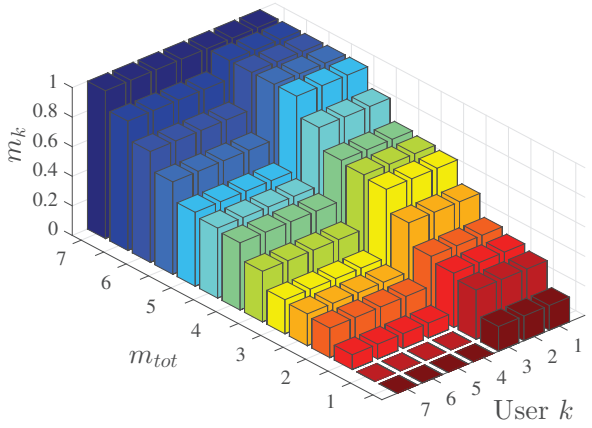

Fig. 4: The optimal memory allocations for $K = 7$, and rates vector $\boldsymbol{C} = [0.2,\ 0.2,\ 0.2,\ 0.5,\ 0.6,\ 0.7,\ 0.7]$.

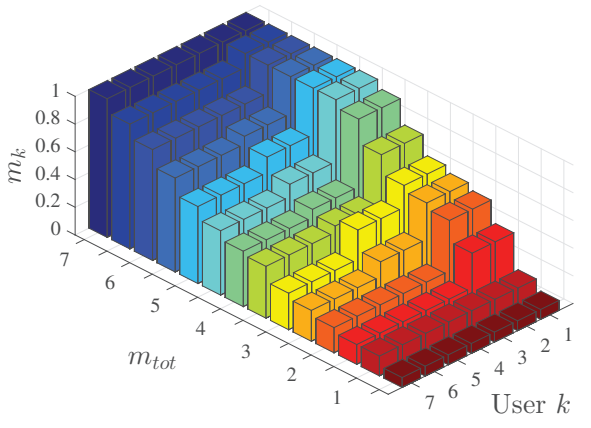

Fig. 5: The optimal memory allocations for $K = 7$, and rates vector $\boldsymbol{C} = [0.2,\ 0.2,\ 0.4,\ 0.4,\ 0.6,\ 0.7,\ 0.8]$.

$[0.2,\ 0.2,\ 0.4,\ 0.4,\ 0.6,\ 0.7,\ 0.8]$, respectively. A general observation is that the optimal memory allocation balances the gain attained from assigning larger memories for users with weak links and the multicast gain achieved by equating the cache sizes. Consequently, in the optimal memory allocation the users are divided into groups according to their rates, the groups that include users with low rates are assigned larger fractions of the cache memory budget, and users within each group are given equal cache sizes.

These characteristics are illustrated in Fig 4, which shows that the users are grouped into $\mathcal{G}_1 = \{1, 2, 3\}$ and $\mathcal{G}_2 = \{4, 5, 6, 7\}$ for all $m_{\text{tot}} \in [0, 7)$. For example, for $m_{\text{tot}} = 2$, we have $\boldsymbol{m}^* = [0.4,\ 0.4,\ 0.4,\ 0.2,\ 0.2,\ 0.2,\ 0.2]$. On the other hand, Fig 5 shows that the users grouping not only depend on the rates $\boldsymbol{C}$, but also on the cache memory budget m_{tot}. For instance, for $m_{\text{tot}} = 2$, we have $\boldsymbol{m}^* = [0.5455,\ 0.5455,\ 0.1818,\ 0.1818,\ 0.1818,\ 0.1818,\ 0.1818]$, while in the case $m_{\text{tot}} = 3$, the allocation becomes $\boldsymbol{m}^* = [0.6316,\ 0.6316,\ 0.4737,\ 0.4737,\ 0.2632,\ 0.2632,\ 0.2632]$, i.e., for $m_{\text{tot}} = 2$, we have two groups $\mathcal{G}_1 = \{1, 2\}$ and $\mathcal{G}_2 = \{3, 4, 5, 6, 7\}$, however, for $m_{\text{tot}} = 3$, we have three groups $\mathcal{G}_1 = \{1, 2\}$, $\mathcal{G}_2 = \{3, 4\}$, and $\mathcal{G}_2 = \{5, 6, 7\}$.

VII. Conclusions

In this paper, we have considered centralized coded caching system where the links between the server and the users have fixed and unequal capacities. The server not only designs the users' cache contents, but also assigns their cache memory sizes subject to a cache memory budget. We have formulated an optimization problem for minimizing the worst-case delivery completion time (DCT) by jointly optimizing the memory allocation and caching scheme. We have characterized explicitly the optimal cache sizes and caching scheme for the case where the cache memory budget is smaller than or equal to the library size. In particular, the optimal memory allocation is to allocate the cache memory budget uniformly over the users with the q lowest link capacities. That is the solution balances the multicast gain and the gain achieved by transmitting less data to users with low rates. Additionally, we have shown that the solution obtained from the optimization problem outperforms the MaddahAli-Niesen caching scheme [1] under uniform memory allocation.

Future directions include systems with multiple servers storing distinct libraries and total memory budget at the end users that can be partitioned over the libraries.

Appendix A
Proof of Theorem 1

In order to characterize explicitly the optimal memory allocation and caching scheme for $m_{\text{tot}} \leq 1$ and $C_1 \leq \cdots \leq C_K$, we first quantify the optimal caching scheme for a given memory allocation with $\sum_{k=1}^{K} m_k \leq 1$, as follows.

Lemma 2. *For $C_1 \leq \cdots \leq C_K$ and memory allocation $\boldsymbol{m}$ satisfying $\sum_{k=1}^{K} m_k \leq 1$, the optimal caching scheme for (1) is given by $a^*_{\{j\}} = m_j$, $v^*_{\{i,j\}} = u^{*\{i,j\}}_{\{i\}} = u^{*\{i,j\}}_{\{j\}} = \min\{a^*_{\{i\}}, a^*_{\{j\}}\}$, and $v^*_{\{j\}} = 1 - m_j - \sum_{i=1, i\neq j}^{K} \min\{m_i, m_j\}$.*

Proof. By dividing (11) by C_k and summing over k, we get

$$\sum_{k=1}^{K} \sum_{\mathcal{T}\in 2^{[K]}-\{\}:k\in\mathcal{T}} \frac{v_{\mathcal{T}}}{C_k} \geq \sum_{k=1}^{K} \frac{1-m_k}{C_k},$$

$$\Rightarrow \sum_{k=1}^{K} \frac{v_{\{k\}}}{C_k} \geq \sum_{k=1}^{K} \frac{1-m_k}{C_k} - \sum_{\mathcal{T}\in 2^{[K]}-\{\}:|\mathcal{T}|\geq 2} \sum_{j\in\mathcal{T}} \frac{v_{\mathcal{T}}}{C_j}.$$

Therefore, we get the lower bound

$$\Theta_{\mathfrak{A},\mathfrak{D}} \geq \sum_{k=1}^{K} \frac{1-m_k}{C_k} - \sum_{\mathcal{T}\in 2^{[K]}-\{\}:|\mathcal{T}|\geq 2} v_{\mathcal{T}} \left(\frac{-1}{\min_{i\in\mathcal{T}} C_i} + \sum_{j\in\mathcal{T}} \frac{1}{C_j} \right).$$

Additionally, for $C_1 \leq \cdots \leq C_K$, we have

$$\Theta_{\mathfrak{A},\mathfrak{D}} \geq \sum_{k=1}^{K} \frac{1-m_k}{C_k} - \sum_{i=1}^{K-1} \sum_{j=i+1}^{K} \frac{1}{C_j} \sum_{\mathcal{T}\in 2^{[K]}-\{\}:\{i,j\}\subset\mathcal{T}} v_{\mathcal{T}},$$

$$\geq \sum_{k=1}^{K} \frac{1-m_k}{C_k} - \sum_{i=1}^{K-1} \sum_{j=i+1}^{K} \frac{\min\{m_i, m_j\}}{C_j}.$$

where the last inequality follows from the fact that the multicast transmissions that include users $\{i,j\}$ are limited by the side information stored at each of them, which is upper bounded by the cache memory size, i.e., $\sum_{\mathcal{T}\in 2^{[K]}-\{\}:\{i,j\}\subset\mathcal{T}} v_{\mathcal{T}} \leq \min\{m_i,m_j\}$. Moreover, for $\sum_{i=1}^{K} m_i \leq 1$, the lower bound is achieved by setting $a_{\{j\}}=m_j$, $v_{\{j\}}=1-m_j-\sum_{i=1,i\neq j}^{K}\min\{m_i,m_j\}$, and $v_{\{i,j\}}=u_{\{i\}}^{\{i,j\}}=u_{\{j\}}^{\{i,j\}}=\min\{a_{\{i\}},a_{\{j\}}\}$. □

Now, using Lemma 2, we can simplify (1) to

$$\min_{\boldsymbol{m}} \quad \sum_{k=1}^{K}\frac{1-m_k}{C_k}-\sum_{i=1}^{K-1}\sum_{j=i+1}^{K}\frac{\min\{m_i,m_j\}}{C_j} \tag{14a}$$

$$\text{subject to} \quad \sum_{k=1}^{K} m_k \leq m_{\text{tot}}, \tag{14b}$$

$$0\leq m_k\leq 1,\ \forall\, k\in[K]. \tag{14c}$$

Next, we show that the optimal memory allocation obtained from (14) satisfies $m_1^*\geq m_2^*\geq\cdots\geq m_K^*$.

Lemma 3. *For $C_1\leq\cdots\leq C_K$ and $m_{tot}\leq 1$, the objective function of (14) satisfies $\Theta_{\mathfrak{A},\mathfrak{D}}(\boldsymbol{m})\leq\Theta_{\mathfrak{A},\mathfrak{D}}(\tilde{\boldsymbol{m}})$, where $m_i=\tilde{m}_i$, for $i\in[K]-\{r,r+1\}$, and some $r\in[K-1]$. Additionally, $m_r=\tilde{m}_{r+1}=\alpha+\delta$, $m_{r+1}=\tilde{m}_r=\alpha$, for $\delta,\alpha\geq 0$, and $m_1\geq m_2\geq\cdots\geq m_r$.*

Proof. For $\boldsymbol{m}=[m_1,m_2,\ldots,m_{r-1},\alpha+\delta,\alpha,m_{r+2},\ldots,m_K]$ and $\tilde{\boldsymbol{m}}=[m_1,m_2,\ldots,m_{r-1},\alpha,\alpha+\delta,m_{r+2},\ldots,m_K]$, we have $\Theta_{\mathfrak{A},\mathfrak{D}}(\boldsymbol{m})-\Theta_{\mathfrak{A},\mathfrak{D}}(\tilde{\boldsymbol{m}})=\chi_1+\chi_2$, where

$$\begin{aligned}\chi_1&=\frac{1-m_r}{C_r}+\frac{1-m_{r+1}}{C_{r+1}}-\frac{1-\tilde{m}_r}{C_r}-\frac{1-\tilde{m}_{r+1}}{C_{r+1}}\\&=\delta\left(\frac{1}{C_{r+1}}-\frac{1}{C_r}\right).\end{aligned}$$

$$\begin{aligned}\chi_2&=\sum_{i=1}^{r-1}\left(\frac{\min\{m_i,\tilde{m}_r\}}{C_r}+\frac{\min\{m_i,\tilde{m}_{r+1}\}}{C_{r+1}}\right)\\&\quad-\sum_{i=1}^{r-1}\left(\frac{\min\{m_i,m_r\}}{C_r}+\frac{\min\{m_i,m_{r+1}\}}{C_{r+1}}\right)\\&=\left(\frac{1}{C_{r+1}}-\frac{1}{C_r}\right)\sum_{i=1}^{r-1}\left(\min\{m_i,\alpha+\delta\}-\min\{m_i,\alpha\}\right)\\&=\delta(r-1)\left(\frac{1}{C_{r+1}}-\frac{1}{C_r}\right).\end{aligned}$$

Thus, $\chi_1+\chi_2=r\delta\left(\frac{1}{C_{r+1}}-\frac{1}{C_r}\right)\leq 0$, as $C_{r+1}\geq C_r$. □

Using Lemma 3, (14) can be simplified to (15).

Lemma 4. *For $C_1\leq\cdots\leq C_K$ and $m_{tot}\leq 1$, optimization problem (1) reduces to*

$$\min_{\boldsymbol{m}} \quad \sum_{k=1}^{K}\frac{1-k\,m_k}{C_k} \tag{15a}$$

$$\textit{subject to} \quad \sum_{k=1}^{K} m_k\leq m_{tot}, \tag{15b}$$

$$m_{k+1}\leq m_k,\ \forall\, k\in[K-1], \tag{15c}$$

$$m_k\geq 0,\ \forall\, k\in[K]. \tag{15d}$$

Moreover, the optimal memory allocation for (15), can be obtained from the following maximization problem.

$$\max_{\boldsymbol{m}\succeq 0} \quad \sum_{k=1}^{K}\frac{k\,m_k}{C_k} \tag{16a}$$

$$\text{subject to}\begin{bmatrix}1&1&1&\cdots&1&1\\-1&1&0&\cdots&0&0\\0&-1&1&\cdots&0&0\\\vdots&\vdots&&\vdots&\vdots&\vdots\\0&0&0&0&-1&1\end{bmatrix}\begin{bmatrix}m_1\\m_2\\m_3\\\vdots\\m_K\end{bmatrix}\leq\begin{bmatrix}m_{\text{tot}}\\0\\0\\\vdots\\0\end{bmatrix}. \tag{16b}$$

Finally, the optimal memory allocation in Theorem 1 is obtained by solving the dual of the linear program in (16).

Appendix B
Proof of Lemma 1

First, we distribute m_{tot} uniformly over the users, i.e., $m_j=m_{\text{tot}}/K,\forall j\in[K]$. In the placement phase, each user caches $\binom{K-1}{m_{\text{tot}}-1}$ pieces from each file, and the size of each piece is $F/\binom{K}{m_{\text{tot}}}$ bits. Therefore, the placement scheme is described by $a_{\mathcal{S}}=1/\binom{K}{m_{\text{tot}}}$ for $\mathcal{S}\in\left\{\mathcal{S}\in 2^{[K]}:|\mathcal{S}|=m_{\text{tot}}\right\}$. On the other hand, the delivery phase is defined by $v_{\mathcal{T}}=1/\binom{K}{m_{\text{tot}}}$ for $|\mathcal{T}|=m_{\text{tot}}+1$. Additionally, $u_{\mathcal{S}}^{\mathcal{T}}=v_{\mathcal{T}}$ for $\mathcal{S}\in\{\mathcal{T}-\{j\}:j\in\mathcal{T}\}$. In turn, the DCT under this scheme is given by

$$\begin{aligned}\Theta_{\text{unif}}(m_{\text{tot}},\boldsymbol{C})&=\sum_{\mathcal{T}\in 2^{[K]}-\{\}}\frac{v_{\mathcal{T}}}{\min_{j\in\mathcal{T}}C_j},\\&=\sum_{\mathcal{T}\in 2^{[K]}-\{\}:\ |\mathcal{T}|=m_{\text{tot}}+1}\frac{1/\binom{K}{m_{\text{tot}}}}{\min_{j\in\mathcal{T}}C_j}=\frac{1}{\binom{K}{m_{\text{tot}}}}\sum_{j=1}^{K-m_{\text{tot}}}\frac{\binom{K-j}{m_{\text{tot}}}}{C_j},\end{aligned}$$

since $C_1\leq C_2\leq\cdots\leq C_K$ and there are $\binom{K-j}{m_{\text{tot}}}$ sets of size $m_{\text{tot}}+1$ that include user j and do not include users $\{1,2,\ldots,j-1\}$.

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:1610.03792*, 2016.

[5] R. Timo and M. Wigger, "Joint cache-channel coding over erasure broadcast channels," in *IEEE ISWCS*, 2015, pp. 201–205.

[6] S. S. Bidokhti, M. Wigger, and R. Timo, "Noisy broadcast networks with receiver caching," *arXiv:1605.02317*, 2016.

[7] S. S. Bidokhti, M. Wigger, and A. Yener, "Gaussian broadcast channels with receiver cache assignment," *IEEE ICC*, 2017.

[8] A. M. Ibrahim, A. A. Zewail, and A. Yener, "Centralized coded caching with heterogeneous cache sizes," *IEEE WCNC*, 2017.

[9] A. F. Dana, R. Gowaikar, R. Palanki, B. Hassibi, and M. Effros, "Capacity of wireless erasure networks," *IEEE Trans. Info. Theory*, vol. 52, no. 3, pp. 789–804, Mar. 2006.

[10] Q. Yu, M. A. Maddah-Ali, and A. S. Avestimehr, "The exact rate-memory tradeoff for caching with uncoded prefetching," *arXiv:1609.07817*, 2016.