On Coded Caching with Heterogeneous Distortion Requirements

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

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

Abstract—**This paper considers heterogeneous coded caching where the users have unequal distortion requirements. The server is connected to the users via an error-free multicast link and designs the users' cache sizes subject to a total memory budget. In particular, in the placement phase, the server jointly designs the users' cache sizes and the cache contents. To serve the users' requests, in the delivery phase, the server transmits signals that satisfy the users' distortion requirements. An optimization problem with the objective of minimizing the worst-case delivery load subject to the total cache memory budget and users' distortion requirements is formulated. The optimal solution for uncoded placement and linear delivery is characterized explicitly and is shown to exhibit a threshold policy with respect to the total cache memory budget. As a byproduct of the study, a caching scheme for systems with fixed cache sizes that outperforms the state-of-art is presented.**

I. INTRODUCTION

Wireless data traffic is increasing at an unprecedented rate due to the demand on video streaming services, which have accounted for 60 percent of total mobile data traffic in 2016 [1]. Efficient utilization of network resources is essential in order to accommodate the growth in data traffic. Caching utilizes the cache memories in the network nodes to shift some of the data traffic to off-peak hours. Reference [2] has proposed *coded caching*, where the end-users' cache contents are designed in a way to enable subsequently serving the users using multicast transmissions, reducing the delivery load on the server.

The significant gain achieved by coded caching has motivated studying its fundamental limits in various setups [2]–[28]. For instance, references [7]–[11] have studied coded caching in multi-hop networks. Coded caching for device-to-device (D2D) networks has been studied in [12]–[14]. Caching with security requirements have been studied in [9], [15]–[17].

Content delivery networks consists of heterogeneous end-devices that have varying storage, computational capabilities, and viewing preferences. In turn, the effect of heterogeneity in cache sizes on the delivery load has been studied in [14], [18]–[22]. Additionally, optimizing the users' cache sizes subject to a network-wide total memory budget has been considered in [23]–[26].

The heterogeneity in users' preferences for content motivates developing coded caching schemes with different quality-of-service requirements per user. In this realm, coded caching schemes with heterogeneous distortion requirements have been studied in references [27]–[29]. In particular, reference [27] has considered a centralized system where files are modeled as independent and identically distributed (i.i.d.) samples from a Gaussian source. Each file is represented by a number of layers equal to the number of users in the system, and accessing the first k layers guarantees that the kth user's distortion requirement is satisfied. In this setup, reference [27] has minimized the squared error distortion for given delivery load, cache sizes, and popularity distribution. Reference [28] has studied the problem of minimizing the delivery load in a centralized caching system with heterogeneous distortion requirements and heterogeneous cache sizes at the users. In particular, reference [28] has considered a separation approach where the memory allocation over the layers and the caching scheme are optimized separately.

In this work, like reference [28], we study the problem of minimizing the delivery load given heterogeneous distortion requirements at the users. Different from references [27] and [28], we consider that the server not only designs the users' cache contents, but also optimizes their cache sizes subject to a total cache memory budget. In contrast to [28], we jointly design the users cache sizes, the memory allocation over the layers, and the caching scheme in order to minimize the delivery load that achieves certain distortion requirements at the users. Under uncoded placement and linear delivery, we show that the joint optimization problem reduces to a memory allocation problem over the layers which can be solved analytically. In particular, the memory allocation over the layers is obtained using a threshold policy, which depends on the available total cache memory budget and the target distortion requirements at the users. We extend the cut-set bound in [28] to systems with total cache memory budget and compare it with the delivery load achieved by the proposed scheme. We observe that the cut-set bound is achievable for large total cache memory budgets.

Although our primary goal in this study is to demonstrate the merit of optimally allocating the cache sizes at different users, we note that the new caching scheme we propose improves on the caching schemes in [28] for systems with fixed cache sizes, since we jointly optimize the caching scheme
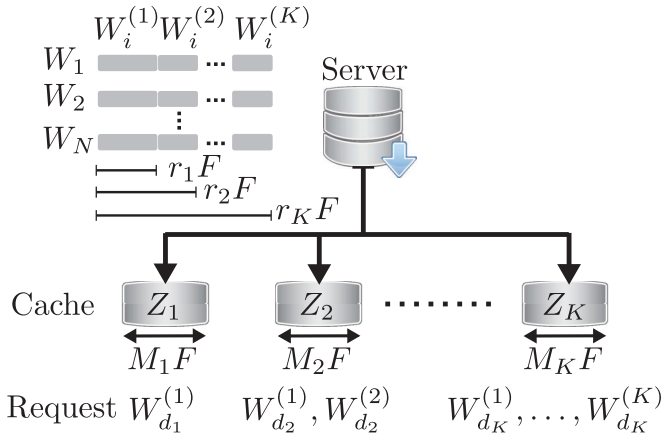
Fig. 1. Caching system with heterogeneous distortion requirements $D_1 \geq D_2 \geq \cdots \geq D_K$.

and the memory allocation over the layers. More specifically, the flexibility in our scheme allows us to exploit the multicast opportunities over all layers. Our numerical results confirm the gain attained by our proposed scheme over the two caching schemes presented in [28]. Finally, we present a numerical example that shows the suboptimality of exploiting only the intra-layer multicast opportunities without taking into account the inter-layer multicast opportunities.

Notation: Vectors are represented by boldface letters, $\oplus$ refers to bitwise XOR operation, $|W|$ denotes size of W, $\mathcal{A} \setminus \mathcal{B}$ denotes the set of elements in $\mathcal{A}$ and not in $\mathcal{B}$, $[K] \triangleq \{1, \ldots, K\}$, ϕ denotes the empty set, $\subsetneq_\phi [K]$ denotes non-empty subsets of $[K]$.

II. SYSTEM MODEL

We consider a server connected to K users via an error-free multicast network, see Fig. 1. The server has a library of N files, $W_1, \ldots, W_N$, each file W_j consists of F i.i.d. samples $[S_{j,1}, \ldots, S_{j,F}]$ from a uniform discrete source. That is, $S_{j,i}$ is uniformly distributed over $\mathbb{F}_q$, $Pr(S_{j,i} = s) = 1/q$ for $s \in \mathbb{F}_q$. We use Hamming distortion as our distortion measure, i.e., we have

$$d(s, \hat{s}) = \begin{cases} 0, & \text{if } s = \hat{s}, \\ 1, & \text{if } s \neq \hat{s}. \end{cases} \tag{1}$$

The rate-distortion function for a uniform discrete source with Hamming distortion is given as [30]

$$\rho(D) = \log(q) - H(D) - D\log(q - 1), \tag{2}$$

where $0 \leq D \leq 1 - 1/q$. Since a uniform discrete source with Hamming distortion is successfully refinable [31], each file can be represented using a scalable layered description that achieves the target rate $r_l \triangleq \rho(D_l)$ at layer l for distortion requirements $D_1 \geq D_2 \geq \cdots \geq D_K$. In particular, layer 1 of each file is a coarse description with size $r_1 F$ bits, while layer $l \in \{2, \ldots, K\}$ is a refinement with size $(r_l - r_{l-1})F$ bits. In turn, we represent file W_j by $W_j^{(1)}, \ldots, W_j^{(K)}$, where $W_j^{(l)}$ denotes layer l of file j. Additionally, we have that D_k

is the distortion requirement at user k, i.e., user k needs layers $\{1, \ldots, k\}$ in order to decode the requested file. Consequently, we define the target rate vector $\boldsymbol{r} = [r_1, .., r_K]$.

The sizes of the users' cache memories are determined by the server. In particular, the server allocates $M_k F$ bits to user k such that $\sum_{k=1}^K M_k F \leq m_{\text{tot}} N F$ bits, where m_{tot} is the cache memory budget normalized by 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 consider the regime where the number of files is greater than or equal the number of users, i.e., $N \geq K$, and $M_k \in [0, Nr_k]$, $\forall k \in [K]$ which implies $m_k \in [0, r_k]$, $\forall k \in [K]$. We denote the memory size vector by $\boldsymbol{M} = [M_1, \ldots, M_K]$ and its normalized version by $\boldsymbol{m} = [m_1, \ldots, m_K]$.

The system has two operational phases: placement and delivery [26]. In the placement phase, the server assigns the users' cache sizes and the contents of the users' cache memories subject to the distortion requirements $\boldsymbol{D} = [D_1, \ldots, D_K]$ and the cache memory budget m_{tot}, without the knowledge of the users' demands. In particular, the server places a subset of the library, Z_k, at the cache memory of user k, such that $|Z_k| \leq M_k F$ bits. In the delivery phase, user k requests the file W_{d_k} which must be recovered with an average distortion less than or equal to D_k. Equivalently, user k requests layers $[k]$ from file W_{d_k}. The requested files are represented by the demand vector $\boldsymbol{d} = [d_1, \ldots, d_K]$. We assume that $\boldsymbol{d}$ consists of identical and independent uniform random variables over the files [2]. In order to guarantee the desired distortion requirements at the users, the server needs to deliver the bits that are not cached by the users in each requested layer. In particular, the server sends the signals $X_{\mathcal{T}, \boldsymbol{d}}$ to the users in the sets $\mathcal{T} \subsetneq_\phi [K]$. User k should be able to decode $W_{d_k}^{(1)}, \ldots, W_{d_k}^{(k)}$ by utilizing the cached contents Z_k and the transmitted signals $X_{\mathcal{T}, \boldsymbol{d}}, \mathcal{T} \subsetneq_\phi [K]$. Formally, we have the following definitions.

Definition 1. *For a given file size F, a caching scheme is defined by the collection of cache placement, encoding, and decoding functions $(\varphi_k(.), \psi_{\mathcal{T}, \boldsymbol{d}}(.), \mu_{\boldsymbol{d}, k}(.))$. A cache placement function*

$$\varphi_k : \mathbb{F}_q^F \times \cdots \times \mathbb{F}_q^F \to [2^{M_k F}], \tag{3}$$

maps the N files to the cache of user k, i.e., $Z_k = \varphi_k(W_1, W_2, .., W_N)$. Given a demand vector $\boldsymbol{d}$, an encoding function

$$\psi_{\mathcal{T}, \boldsymbol{d}} : \mathbb{F}_q^F \times \cdots \times \mathbb{F}_q^F \to [2^{v_{\mathcal{T}} F}], \tag{4}$$

maps the requested files to a signal with length $v_{\mathcal{T}} F$ bits, which is sent to the users in $\mathcal{T}$, i.e., $X_{\mathcal{T}, \boldsymbol{d}} = \psi_{\mathcal{T}, \boldsymbol{d}}(W_{d_1}, .., W_{d_K})$. Finally, a decoding function

$$\mu_{\boldsymbol{d}, k} : [2^{M_k F}] \times [2^{RF}] \to \mathbb{F}_q^F, \tag{5}$$

with $R \triangleq \sum_{\mathcal{T} \subsetneq_\phi [K]} v_{\mathcal{T}}$, maps Z_k and $X_{\mathcal{T}, \boldsymbol{d}}, \mathcal{T} \subsetneq_\phi [K]$ to $\hat{W}_{d_k}$, i.e., $\hat{W}_{d_k} = \mu_{\boldsymbol{d}, k}\left(X_{\{1\}, \boldsymbol{d}}, X_{\{2\}, \boldsymbol{d}}, \ldots, X_{[K], \boldsymbol{d}}, Z_k\right)$. ∎

Definition 2. *For given normalized cache sizes $\boldsymbol{m}$, distortion*

requirements $D_1 \geq D_2 \geq \cdots \geq D_K$, and $r_l = \rho(D_l)$, the delivery load $R(\boldsymbol{m}, \boldsymbol{r})$ is achievable if there exists a sequence of caching schemes such that $\lim_{F \to \infty} \frac{1}{F} \sum_{i=1}^{F} d(S_{d_k,i}, \hat{S}_{d_k,i}) \leq D_k$, $\forall k \in [K]$, $\forall d_k \in [N]$. Furthermore, the infimum over all achievable delivery loads is denoted by $R^*(\boldsymbol{m}, \boldsymbol{r})$. ∎

In this work, we consider the class of cache placement schemes, $\mathfrak{A}$, in which user k cache uncoded pieces of layers $[k]$ of the files, and the class of delivery schemes, $\mathfrak{D}$, where the multicast signals are formed using linear codes. Note that the uniform demands assumption implies that each user should cache the same number of bits from all files, i.e., user k dedicates $m_k F$ bits to each file.

Definition 3. *For given normalized cache sizes $\boldsymbol{m}$ and target rates $\boldsymbol{r}$, the worst-case delivery load under an uncoded placement scheme in $\mathfrak{A}$, and a linear delivery policy in $\mathfrak{D}$, is defined as*

$$R_{\mathfrak{A}, \mathfrak{D}}(\boldsymbol{m}, \boldsymbol{r}) \triangleq \max_{\boldsymbol{d} \in [N]^K} R_{\boldsymbol{d}, \mathfrak{A}, \mathfrak{D}} = \sum_{\mathcal{T} \subsetneq_\phi [K]} v_\mathcal{T}. \quad (6)$$

Furthermore, by taking the infimum over $\mathfrak{A}$ and all possible delivery policies, we get $R^*_{\mathfrak{A}}(\boldsymbol{m}, \boldsymbol{r})$. ∎

The trade-off between the delivery load and the total cache memory budget is defined as follows.

Definition 4. *For given normalized cache memory budget m_{tot} and target rate vector $\boldsymbol{r}$, the infimum over all achievable worst-case delivery loads is given as*

$$R^*(m_{tot}, \boldsymbol{r}) = \inf_{\boldsymbol{m} \in \mathcal{M}(m_{tot}, \boldsymbol{r})} R^*(\boldsymbol{m}, \boldsymbol{r}), \quad (7)$$

where $\mathcal{M}(m_{tot}, \boldsymbol{r}) = \left\{ \boldsymbol{m} \middle| 0 \leq m_k \leq r_k, \sum_{k=1}^{K} m_k = m_{tot} \right\}$. Furthermore, we have

$$R^*_{\mathfrak{A}}(m_{tot}, \boldsymbol{r}) = \inf_{\boldsymbol{m} \in \mathcal{M}(m_{tot}, \boldsymbol{r})} R^*_{\mathfrak{A}}(\boldsymbol{m}, \boldsymbol{r}). \quad (8)$$

∎

III. A NOVEL CACHING SCHEME

Given the layered description of the files explained in Section II, the problem of designing the users' cache contents, under the uncoded placement assumption, can be decomposed into K placement problems, each of which corresponds to one of the layers [28]. In particular, the cache memory at user k is partitioned over layers $[k]$ and $m_k^{(l)}$ denotes the normalized cache memory dedicated to layer l at user k. In turn, the placement problem of layer l is equivalent to the placement problem with $K - l + 1$ users and unequal cache memories $\boldsymbol{m}^{(l)} = [m_l^{(l)}, \ldots, m_K^{(l)}]$ addressed in [22], [26]. By contrast, in the delivery phase, decoupling over the layers is suboptimal. That is, we need to jointly design the multicast signals over all layers, in order to utilize all multicast opportunities. Next, we explain the cache placement and delivery schemes in detail.

A. Placement Phase

Under uncoded placement, the placement phase is decomposed into K layers. In particular, layer l of each file is partitioned over users $\{l, \ldots, K\}$. That is, $W_j^{(l)}$ is divided into subfiles, $\tilde{W}_{j,\mathcal{S}}^{(l)}, \mathcal{S} \subset \{l, \ldots, K\}$, which are labeled by the set of users $\mathcal{S}$ exclusively storing them. We assume that $|\tilde{W}_{j,\mathcal{S}}^{(l)}| = a_\mathcal{S}^{(l)} F$ bits $\forall j \in [N]$, i.e., we have symmetric partitioning over the files and the *allocation variable* $a_\mathcal{S}^{(l)} \in [0, r_l - r_{l-1}]$ defines the size of $\tilde{W}_{j,\mathcal{S}}^{(l)}$. In turn, the set of feasible placement schemes for layer l is defined by

$$\mathfrak{A}^{(l)}(\boldsymbol{m}^{(l)}, \boldsymbol{r}) = \left\{ \boldsymbol{a}^{(l)} \middle| \sum_{\mathcal{S} \subset \{l, \ldots, K\}} a_\mathcal{S}^{(l)} = r_l - r_{l-1}, \right.$$
$$\left. \sum_{\mathcal{S} \subset \{l, \ldots, K\}: k \in \mathcal{S}} a_\mathcal{S}^{(l)} \leq m_k^{(l)}, \forall k \in \{l, \ldots, K\} \right\}, \quad (9)$$

where $\boldsymbol{a}^{(l)}$ is the vector representation of $\{a_\mathcal{S}^{(l)}\}_\mathcal{S}$. Note that the first constraint follows from the fact that the lth layer of each file is partitioned over the sets $\mathcal{S} \subset \{l, \ldots, K\}$, while the second constraint represents the cache size constraint at layer l for user k. Therefore, the cache content placed at user k is given by

$$Z_k = \bigcup_{l \in [k]} \bigcup_{j \in [N]} \bigcup_{\mathcal{S} \subset \{l, \ldots, K\}: k \in \mathcal{S}} \tilde{W}_{j,\mathcal{S}}^{(l)}. \quad (10)$$

B. Delivery Phase

In order to deliver the missing subfiles, the server sends the sequence of unicast/multicast signals $X_{\mathcal{T}, \boldsymbol{d}}$, $\mathcal{T} \subset [K]$. In particular, the multicast signal to the users in $\mathcal{T}$ is defined by

$$X_{\mathcal{T}, \boldsymbol{d}} = \oplus_{j \in \mathcal{T}} \bigcup_{l=1}^{L} W_{d_j}^{(l), \mathcal{T}}, \quad (11)$$

where $L \triangleq \min_{i \in \mathcal{T}} i$ and $W_{d_j}^{(l), \mathcal{T}}$ denotes a subset of $W_{d_j}^{(l)}$ which is delivered to user j via $X_{\mathcal{T}, \boldsymbol{d}}$. $W_{d_j}^{(l), \mathcal{T}}$ is constructed using the side-information available at the users in $\mathcal{T} \setminus \{j\}$. That is, if $W_{d_j, \mathcal{S}}^{(l), \mathcal{T}}$ denotes the subset of $W_{d_j}^{(l), \mathcal{T}}$ cached by the users in $\mathcal{S}$, then we have

$$W_{d_j}^{(l), \mathcal{T}} = \bigcup_{\mathcal{S} \subset \{l, \ldots, K\} \setminus \{j\}: \mathcal{T} \setminus \{j\} \subset \mathcal{S}} W_{d_j, \mathcal{S}}^{(l), \mathcal{T}}. \quad (12)$$

Additionally, we denote $|X_{\mathcal{T}, \boldsymbol{d}}| = \sum_{l=1}^{L} |W_{d_j}^{(l), \mathcal{T}}| = v_\mathcal{T} F$ bits $\forall j \in \mathcal{T}$ and $|W_{d_j, \mathcal{S}}^{(l), \mathcal{T}}| = u_\mathcal{S}^{(l), \mathcal{T}} F$ bits. That is, the transmission variable $v_\mathcal{T} \in [0, r_L]$ and the assignment variable $u_\mathcal{S}^{(l), \mathcal{T}} \in [0, a_\mathcal{S}^{(l)}]$ determine the structure of the signal $X_{\mathcal{T}, \boldsymbol{d}}$. Furthermore, the unicast signal $X_{\{k\}, \boldsymbol{d}}$ delivers the missing pieces of $\bigcup_{l=1}^{k} W_{d_k}^{(l)}$ to user k, i.e., the pieces that had not been delivered by multicast signals and are not cached by user k. We assume $|X_{\{k\}, \boldsymbol{d}}| = \sum_{l=1}^{k} v_{\{k\}}^{(l)} F$ bits.

Next, we explain that all linear delivery schemes under uncoded placement can be described by the following linear constraints on the transmission and assignment variables.

In particular, for given allocation $\{\boldsymbol{a}^{(l)}\}_l$, the set of linear delivery schemes, $\mathfrak{D}(\boldsymbol{a}^{(1)}, \ldots, \boldsymbol{a}^{(K)}, \boldsymbol{r})$, is defined by

$$v_{\mathcal{T}} = \sum_{l=1}^{L} \sum_{\mathcal{S} \in \mathcal{B}_j^{(l),\mathcal{T}}} u_{\mathcal{S}}^{(l),\mathcal{T}}, \forall \mathcal{T} \subset [K] \text{ s.t. } |\mathcal{T}| \geq 2, \forall j \in \mathcal{T}, \quad (13)$$

$$v_{\{k\}}^{(l)} + \sum_{\mathcal{T} \subset \{l,\ldots,K\}: k \in \mathcal{T}, |\mathcal{T}| \geq 2} \sum_{\mathcal{S} \in \mathcal{B}_k^{(l),\mathcal{T}}} u_{\mathcal{S}}^{(l),\mathcal{T}} + \sum_{\mathcal{S} \subset \{l,\ldots,K\}: k \in \mathcal{S}} a_{\mathcal{S}}^{(l)} \geq$$
$$r_l - r_{l-1}, \forall l \in [K], \; \forall k \in \{l, \ldots, K\}, \quad (14)$$

$$\sum_{\mathcal{T} \subset \{l,\ldots,K\}: j \in \mathcal{T}, \mathcal{T} \cap \mathcal{S} \neq \phi} u_{\mathcal{S}}^{(l),\mathcal{T}} \leq a_{\mathcal{S}}^{(l)}, \forall l \in [K], \forall j \notin \mathcal{S}, \forall \mathcal{S} \in \mathcal{A}^{(l)}, \quad (15)$$

$$0 \leq u_{\mathcal{S}}^{(l),\mathcal{T}} \leq a_{\mathcal{S}}^{(l)}, \forall l \in [K], \forall \mathcal{T} \subsetneq_\phi \{l, \ldots, K\}, \forall \mathcal{S} \in \mathcal{B}^{(l),\mathcal{T}}, \quad (16)$$

where

$$\mathcal{B}_j^{(l),\mathcal{T}} \triangleq \{\mathcal{S} \subset \{l, \ldots, K\} \setminus \{j\} : \mathcal{T} \setminus \{j\} \subset \mathcal{S}\},$$
$$\mathcal{A}^{(l)} \triangleq \{\mathcal{S} \subset \{l, \ldots, K\} : 2 \leq |\mathcal{S}| \leq K - l\},$$

and $\mathcal{B}^{(l),\mathcal{T}} \triangleq \bigcup_{j \in \mathcal{T}} \mathcal{B}_j^{(l),\mathcal{T}}$.

The structural constraints in (13) follows from the structure of the multicast signals in (11) and (12). The delivery completion constraints in (14) guarantee that the unicast and multicast signals complete the lth layer of the requested files, and the redundancy constraints in (15) prevent the transmission of redundant bits to the users. For example, for $K = 3$, the structural constraints are defined as

$$v_{\{1,2\}} = u_{\{2\}}^{(1),\{1,2\}} + u_{\{2,3\}}^{(1),\{1,2\}} = u_{\{1\}}^{(1),\{1,2\}} + u_{\{1,3\}}^{(1),\{1,2\}}, \quad (17a)$$

$$v_{\{1,3\}} = u_{\{3\}}^{(1),\{1,3\}} + u_{\{2,3\}}^{(1),\{1,3\}} = u_{\{1\}}^{(1),\{1,3\}} + u_{\{1,2\}}^{(1),\{1,3\}}, \quad (17b)$$

$$v_{\{2,3\}} = u_{\{3\}}^{(1),\{2,3\}} + u_{\{1,3\}}^{(1),\{2,3\}} + u_{\{3\}}^{(2),\{2,3\}}, \quad (17c)$$

$$= u_{\{2\}}^{(1),\{2,3\}} + u_{\{1,2\}}^{(1),\{2,3\}} + u_{\{2\}}^{(2),\{2,3\}}, \quad (17d)$$

$$v_{\{1,2,3\}} = u_{\{2,3\}}^{(1),\{1,2,3\}} = u_{\{1,3\}}^{(1),\{1,2,3\}} = u_{\{1,2\}}^{(1),\{1,2,3\}}, \quad (17e)$$

the delivery completion constraints for user 3 are defined as

$$v_{\{3\}}^{(1)} + \left(u_{\{1\}}^{(1),\{1,3\}} + u_{\{1,2\}}^{(1),\{1,3\}}\right) + \left(u_{\{2\}}^{(1),\{2,3\}} + u_{\{1,2\}}^{(1),\{2,3\}}\right) +$$
$$u_{\{1,2\}}^{(1),\{1,2,3\}} + \left(a_{\{3\}}^{(1)} + a_{\{1,3\}}^{(1)} + a_{\{2,3\}}^{(1)} + a_{\{1,2,3\}}^{(1)}\right) \geq r_1, \quad (18a)$$

$$v_{\{3\}}^{(2)} + u_{\{2\}}^{(2),\{2,3\}} + a_{\{3\}}^{(2)} + a_{\{2,3\}}^{(2)} \geq r_2 - r_1, \quad (18b)$$

$$v_{\{3\}}^{(3)} + a_{\{3\}}^{(3)} \geq r_3 - r_2, \quad (18c)$$

and the redundancy constraints are defined as

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

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

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

We denote the vector representation of the transmission variables $\{v_{\mathcal{T}}\}_{\mathcal{T}}, \{v_{\{k\}}^{(l)}\}_{k,l}$ and the assignment variables $\{u_{\mathcal{S}}^{(l),\mathcal{T}}\}_{l,\mathcal{T},\mathcal{S}}$ by $\boldsymbol{v}$ and $\boldsymbol{u}$, respectively.

IV. FORMULATION

In this section, we demonstrate that the problem of minimizing the worst-case delivery load by optimizing over the users'

cache sizes, uncoded placement, and linear delivery, can be formulated as a linear program. In particular, given the target rate vector $\boldsymbol{r}$, the total memory budget m_{tot}, and $N \geq K$, the minimum worst-case delivery load under uncoded placement and linear delivery, $R_{\mathfrak{A},\mathfrak{D}}^*(m_{\text{tot}}, \boldsymbol{r})$, is characterized by

O1:
$$\min_{\boldsymbol{a}^{(l)}, \boldsymbol{u}, \boldsymbol{v}, \boldsymbol{m}^{(l)}} \sum_{\mathcal{T} \subsetneq_\phi [K]} v_{\mathcal{T}} \quad (20a)$$

$$\text{subject to} \quad \boldsymbol{a}^{(l)} \in \mathfrak{A}^{(l)}(\boldsymbol{m}^{(l)}, \boldsymbol{r}), \forall l \in [K] \quad (20b)$$

$$(\boldsymbol{u}, \boldsymbol{v}) \in \mathfrak{D}(\boldsymbol{a}^{(1)}, \ldots, \boldsymbol{a}^{(K)}, \boldsymbol{r}), \quad (20c)$$

$$\sum_{k=1}^{K} \sum_{l=1}^{k} m_k^{(l)} = m_{\text{tot}}, \quad (20d)$$

where $\mathfrak{A}^{(l)}(\boldsymbol{m}^{(l)}, \boldsymbol{r})$ is the set of uncoded placement schemes in layer l defined in (9) and $\mathfrak{D}(\boldsymbol{a}^{(1)}, \ldots, \boldsymbol{a}^{(K)}, \boldsymbol{r})$ is the set of feasible linear delivery schemes defined by (13)-(16).

We can also solve the problem of designing the caching scheme and the memory allocation over the layers for systems with fixed cache sizes as in [28]. In particular, for fixed cache sizes $\boldsymbol{m}$, the minimum worst-case delivery load under uncoded placement and linear delivery, $R_{\mathfrak{A},\mathfrak{D}}^*(\boldsymbol{m}, \boldsymbol{r})$, is characterized by

O2:
$$\min_{\boldsymbol{a}^{(l)}, \boldsymbol{u}, \boldsymbol{v}, \boldsymbol{m}^{(l)}} \sum_{\mathcal{T} \subsetneq_\phi [K]} v_{\mathcal{T}} \quad (21a)$$

$$\text{subject to} \quad \boldsymbol{a}^{(l)} \in \mathfrak{A}^{(l)}(\boldsymbol{m}^{(l)}, \boldsymbol{r}), \forall l \in [K] \quad (21b)$$

$$(\boldsymbol{u}, \boldsymbol{v}) \in \mathfrak{D}(\boldsymbol{a}^{(1)}, \ldots, \boldsymbol{a}^{(K)}, \boldsymbol{r}), \quad (21c)$$

$$\sum_{l=1}^{k} m_k^{(l)} = m_k, \forall k \in [K]. \quad (21d)$$

In contrast to the formulation in reference [28], observe that in (21), we jointly design the caching scheme and the memory allocation over the layers, and exploit multicast opportunities over different layers.

V. OPTIMAL CACHE ALLOCATION

In this section, we first characterize the solution to the optimization problem in (20), i.e., we find the achievable worst-case delivery load assuming uncoded placement and linear delivery. Then, we present a lower bound on the trade-off between the minimum worst-case delivery load under any caching scheme and the cache memory budget.

Theorem 1. *Given the target rate vector $\boldsymbol{r}$, $N \geq K$, and the total memory budget $m_{tot} = \sum_{l=1}^{K} t_l f_l$, where $f_1 \triangleq r_1$, $f_l \triangleq r_l - r_{l-1}$ for $l > 1$, $t_l \in \{0, 1, \ldots, K - l + 1\}$ and $t_{l+1} \leq t_l \leq t_{l-1}$, the minimum achievable worst-case delivery load under uncoded placement is given by*

$$R_{\mathfrak{A},\mathfrak{D}}^*(m_{tot}, \boldsymbol{r}) = R_{\mathfrak{A}}^*(\boldsymbol{r}, m_{tot}) = \sum_{l=1}^{K} \frac{(K - l + 1) - t_l}{1 + t_l} f_l. \quad (22)$$

Furthermore, for general $m_{tot} \in [0, \sum_{k=1}^{K} r_k]$, $R_{\mathfrak{A},\mathfrak{D}}^*(\boldsymbol{r}, m_{tot})$ is defined by the lower convex envelope of these points. ∎

Proof. In Appendix A, we show that the optimal solution to the optimization problem in (20) achieves the lower convex envelope of the delivery load points in (22). In particular, for $m_{\text{tot}} = \sum_{i=1}^{K} t_i f_i$ where $t_l \in \{0, 1, \ldots, K - l + 1\}$ and $t_{l+1} \leq t_l \leq t_{l-1}$, the optimal memory allocation is defined as $m_k^{(l)} = t_l f_l / (K - l + 1)$ for $k \in \{l, \ldots, K\}$, i.e., the users are assigned equal cache sizes in each layer. In turn, we apply the MaddahAli-Niesen caching scheme in each layer, i.e., in layer l the placement phase is defined by

$$|\tilde{W}_{j,\mathcal{S}}^{(l)}| = \begin{cases} f_l / \binom{K-l+1}{t_l} F, & \text{for } |\mathcal{S}| = t_l, \\ 0, & \text{otherwise.} \end{cases} \quad (23)$$

and the multicast signals are defined by $\oplus_{j \in \mathcal{T}} \tilde{W}_{d_j, \mathcal{T} \setminus \{j\}}^{(l)}$ for $\mathcal{T} \subset \{l, \ldots, K\}$ and $|\mathcal{T}| = t_l + 1$.

In general, any $m_{\text{tot}} \in [0, \sum_{i=1}^{K} r_i]$ can be represented as $\sum_{i=1}^{K} t_i f_i$ where $t_i = x$, for $i = [y]$, $t_y = x - 1 + \alpha$, $t_i = x - 1$, for $i = \{y + 1, \ldots, K - x + 1\}$, $t_i = K - i + 1$, for $i = \{K - x + 2, \ldots, K\}$, for some $x \in [K]$, $y \in [K - x + 1]$ and $0 < \alpha < 1$. In particular, we have

$$x \triangleq \begin{cases} 1, & 0 < m_{\text{tot}} \leq \sum_{i=1}^{K} f_i, \\ \vdots \\ j, & (j{-}1) \sum_{i=1}^{K-j+1} f_i + \sum_{i=K-j+2}^{K} (K{-}i{+}1) f_i < m_{\text{tot}} \leq j \sum_{i=1}^{K-j} f_i \\ & \qquad\qquad + \sum_{i=K-j+1}^{K} (K{-}i{+}1) f_i, \\ \vdots \\ K, & (K{-}1) f_1 + \sum_{i=2}^{K} (K{-}i{+}1) f_i < m_{\text{tot}} \leq \sum_{i=1}^{K} r_i. \end{cases} \quad (24)$$

and for a given x, we have $y = b$ if

$$x \sum_{i=1}^{b-1} f_i + (x{-}1) \sum_{i=b}^{K-x+1} f_i + \sum_{i=K-x+2}^{K} (K{-}i{+}1) f_i < m_{\text{tot}} \leq$$
$$x \sum_{i=1}^{b} f_i + (x{-}1) \sum_{i=b+1}^{K-x+1} f_i + \sum_{i=K-x+2}^{K} (K{-}i{+}1) f_i. \quad (25)$$

In turn, any $m_{\text{tot}} \in [0, \sum_{k=1}^{K} r_k]$ can be represented as

$$m_{\text{tot}} = x \sum_{l=1}^{y-1} f_l + (x - 1 + \alpha) f_y + (x - 1) \sum_{l=y+1}^{K-x+1} f_l +$$
$$\sum_{l=K-x+2}^{K} (K - l + 1) f_l, \quad (26)$$

and the corresponding minimum worst-case delivery load under uncoded placement and linear delivery is given by

$$R_{\mathfrak{A},\mathfrak{D}}^*(m_{\text{tot}}, \boldsymbol{r}) = \sum_{l=1}^{y-1} \frac{K - l - x + 1}{x + 1} f_l + \sum_{l=y+1}^{K-x+1} \frac{K - y - x + 2}{x} f_l +$$
$$\left(\frac{2(K - y) - x + 3}{x + 1} - \frac{(K - y + 2)(x - 1 + \alpha)}{x(x+1)} \right) f_y. \quad (27)$$

$\square$

For example, for $K = 3$, the corner points of the delivery load cache budget trade-off under uncoded placement in (22) are defined as follows

$$R_{\mathfrak{A},\mathfrak{D}}^*(m_{\text{tot}}, \boldsymbol{r}) = \begin{cases} r_1 + r_2 + r_3, & \text{for } m_{\text{tot}} = 0, \\ r_2 + r_3 - r_1, & \text{for } m_{\text{tot}} = r_1, \\ r_1/2 + r_3 - r_2/2, & \text{for } m_{\text{tot}} = r_2, \\ r_1/2 + r_2/2, & \text{for } m_{\text{tot}} = r_3, \\ r_2/2 - r_1/6, & \text{for } m_{\text{tot}} = r_1 + r_3, \\ r_1/3, & \text{for } m_{\text{tot}} = r_2 + r_3, \\ 0, & \text{for } m_{\text{tot}} = r_1 + r_2 + r_3, \end{cases} \quad (28)$$

which are illustrated in Fig. 2, for $\boldsymbol{r} = [0.5, 0.7, 1]$.

Next, we extend the lower bound on the delivery load for systems with fixed cache sizes in [28, Theorem 2] to systems with cache memory budget.

Proposition 1. *(Extension of [28, Theorem 2]) Given $N \geq K$, the target rate vector $\boldsymbol{r}$, and the total memory budget $m_{tot} \in [0, \sum_{k=1}^{K} r_k]$, the infimum over all achievable delivery loads $R^*(m_{tot}, \boldsymbol{r})$ is lower bounded by*

$$\min_{\boldsymbol{m} \in \mathcal{M}(m_{tot}, \boldsymbol{r})} \left\{ \max_{\mathcal{U} \subset [K]} \left\{ \sum_{k \in \mathcal{U}} r_k - \frac{N \sum_{k \in \mathcal{U}} m_k}{\lfloor N/|\mathcal{U}| \rfloor} \right\} \right\}, \quad (29)$$

where $\mathcal{M}(m_{tot}, \boldsymbol{r}) = \left\{ \boldsymbol{m} \big| 0 \leq m_k \leq r_k, \sum_{k=1}^{K} m_k = m_{tot} \right\}$. $\blacksquare$

For the three-user case, the cut-set bound can be simplified as follows.

Corollary 1. *Given $K = 3$, $N \geq 3$, the target rate vector $\boldsymbol{r}$, and the total memory budget $m_{tot} \in [0, \sum_{k=1}^{K} r_k]$, the infimum over all achievable delivery loads $R^*(m_{tot}, \boldsymbol{r})$ is lower bounded by*

$$\max \left\{ \sum_{l=1}^{3} r_l - \frac{N}{\lfloor N/3 \rfloor} m_{tot}, \frac{\lfloor N/2 \rfloor (r_1 + r_2) + N r_3}{N + \lfloor N/2 \rfloor} - \right.$$
$$\left. \frac{N}{N + \lfloor N/2 \rfloor} m_{tot}, \frac{1}{3} \left(\sum_{l=1}^{3} r_l - m_{tot} \right) \right\}. \quad (30)$$

$\blacksquare$

VI. DISCUSSION AND NUMERICAL RESULTS

In this section, we first consider systems with total cache memory budget and present numerical examples that illustrate the optimal solution of (20). Then, we consider systems with fixed cache sizes and illustrate the gain achieved by our caching scheme compared to the schemes proposed in [28].

A. Systems with Total Cache Memory Budget

In Fig. 2, we compare the achievable worst-case delivery load $R_{\mathfrak{A}}^*(m_{\text{tot}}, \boldsymbol{r})$ defined in Theorem 1 with the lower bound on $R^*(m_{\text{tot}}, \boldsymbol{r})$ defined in Proposition 1, for $K = N = 3$ and the target rates $\boldsymbol{r} = [0.5, 0.7, 1]$. In particular, for $K = 3$ the
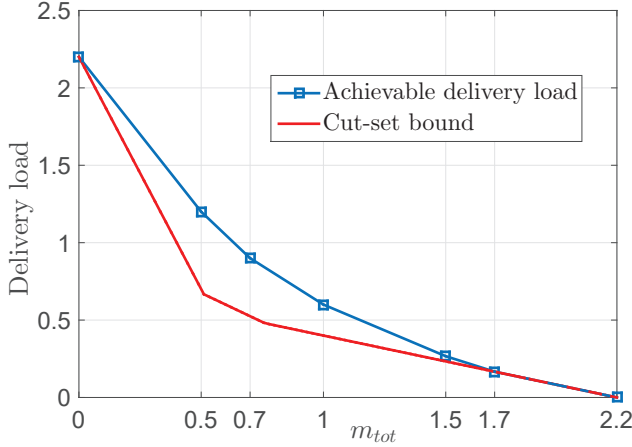
Fig. 2. Comparing the delivery load $R_{\mathfrak{A}}^*(m_{\text{tot}}, \boldsymbol{r})$ in Theorem 1 with the cut-set bound in Proposition 1, for $K = N = 3$, and $\boldsymbol{r} = [0.5, 0.7, 1]$.
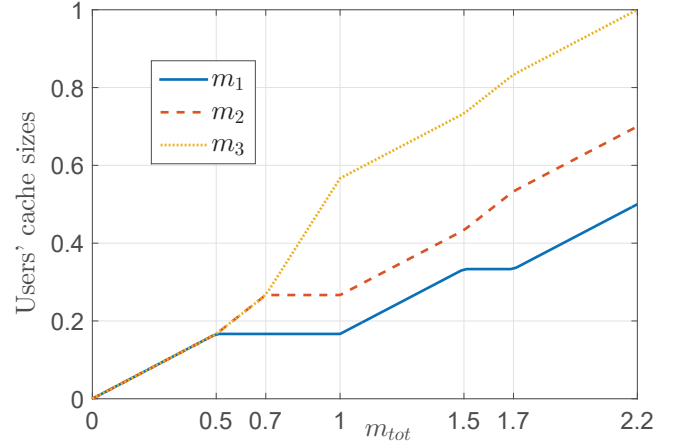
Fig. 3. The memory allocation corresponding to the delivery load in Theorem 1 for $\boldsymbol{r} = [0.5, 0.7, 1]$.

corner points of the achievable delivery load are defined in (28) and the cut-set bound in (30) for $N = 3$ is

$$\max\left\{\sum_{l=1}^{3} r_l - 3m_{\text{tot}}, \frac{r_1 + r_2 + 3r_3 - 3m_{\text{tot}}}{4}, \frac{1}{3}\left(\sum_{l=1}^{3} r_l - m_{\text{tot}}\right)\right\}. \quad (31)$$

Remark 1. *We observe that for large memory budget, $\sum_{l=2}^{K} r_l \leq m_{tot} \leq \sum_{l=1}^{K} r_l$, the cut-set bound can be achieved and the delivery load $R^*(m_{tot}, \boldsymbol{r}) = \frac{1}{K}\left(\sum_{l=1}^{K} r_l - m_{tot}\right)$.* ∎

Recall the notation $f_l = (r_l - r_{l-1})$ and $r_0 = 0$. Fig. 3 shows the optimal allocation for the users' cache sizes that correspond to the achievable delivery load in Fig. 2. From Fig. 3, we observe the optimal memory allocation follows a threshold policy. In particular, we have

- If $m_{\text{tot}} = \alpha f_1$ and $\alpha \in [0,1]$, then $\boldsymbol{m} = [\alpha f_1/3, \alpha f_1/3, \alpha f_1/3]$.
- If $m_{\text{tot}} = f_1 + \alpha f_2$ and $\alpha \in [0,1]$, then $\boldsymbol{m} = [f_1/3, f_1/3 + \alpha f_2/2, f_1/3 + \alpha f_2/2]$.
- If $m_{\text{tot}} = f_1 + f_2 + \alpha f_3$ and $\alpha \in [0,1]$, then $\boldsymbol{m} = [f_1/3, f_1/3 + f_2/2, f_1/3 + f_2/2 + \alpha f_3]$.
- If $m_{\text{tot}} = (1+\alpha)f_1 + f_2 + f_3$ and $\alpha \in [0,1]$, then $\boldsymbol{m} = [(1+\alpha)f_1/3, (1+\alpha)f_1/3 + f_2/2, (1+\alpha)f_1/3 + f_2/2 + f_3]$.
- If $m_{\text{tot}} = 2f_1 + (1+\alpha)f_2 + f_3$ and $\alpha \in [0,1]$, then $\boldsymbol{m} = [2f_1/3, 2f_1/3 + (1+\alpha)f_2/2, 2f_1/3 + (1+\alpha)f_2/2 + f_3]$.
- If $m_{\text{tot}} = (2+\alpha)f_1 + 2f_2 + f_3$ and $\alpha \in [0,1]$, then $\boldsymbol{m} = [(2+\alpha)f_1/3, (2+\alpha)f_1/3 + f_2, (2+\alpha)f_1/3 + f_2 + f_3]$.

B. Systems with Fixed Cache Sizes

For systems with fixed cache sizes, we compare the delivery load achieved by our scheme which performs joint design of the memory allocation over the layers and the caching scheme, with the two separation based schemes in [28]. In reference [28], the memory allocation over the layers follows one of the following heuristic methods:

1) *Proportional cache allocation* (PCA), where user k cache size for layer l is defined as $m_k^{(l)} = f_l m_k / r_k$.

2) *Ordered cache allocation* (OCA), where user k allocates all of its cache to the first b layers if $r_{b-1} < m_k < r_b$.

In turn, in layer l, we have a caching system that consists of $K - l + 1$ users with fixed cache sizes $\boldsymbol{m}^{(l)}$. Reference [28] has proposed designing the caching scheme in each layer separately, i.e., the multicast signals utilize the side-information from one layer. In particular, in layer l, the cache sizes $\boldsymbol{m}^{(l)}$ are further divided into sublayers of equal size and the MaddahAli-Niesen scheme is applied on each sublayer. Additionally, reference [28] has formulated an optimization problem in order to identify the optimal distribution of f_l over the sublayers. We refer to this scheme as the *layered scheme*.

Fig. 4 shows the delivery load obtained from (21), the delivery load achieved by PCA/OCA combined with the layered scheme, and the cut-set bound in [28, Theorem 2], for $K = N = 3$, $\boldsymbol{r} = [0.5, 0.8, 1]$, and $m_k = 0.8\, m_{k+1}$. We observe that the delivery load achieved by (21) is lower than the one achieved by the schemes in [28]. This is attributed to the fact that we not only jointly optimize the memory allocation over the layers and the caching scheme, but also exploit the multicast opportunities over all layers. However, the performance improvement comes at the expense of higher complexity, since the dimension of the optimization problem in (21) grows exponentially with the number of users.

The next example illustrates the suboptimality of exploiting the multicast opportunities in each layer separately.

Example 1. *For $\boldsymbol{m} = [0.1, 0.2, 0.6]$ and $\boldsymbol{r} = [0.2, 0.3, 0.8]$, the optimal solution of (21) is as follows*
Placement phase:

- *Layer 1: $W_j^{(1)}$ is divided into subfiles $\tilde{W}_{j,\{3\}}^{(1)}$ and $\tilde{W}_{j,\{1,2\}}^{(1)}$, such that $a_{\{3\}}^{(1)} = a_{\{1,2\}}^{(1)} = 0.1$.*
- *Layer 2: $W_j^{(2)}$ is stored at user 2, i.e., $a_{\{2\}}^{(2)} = 0.1$.*
- *Layer 3: $W_j^{(3)}$ is stored at user 3, i.e., $a_{\{3\}}^{(3)} = 0.5$.*

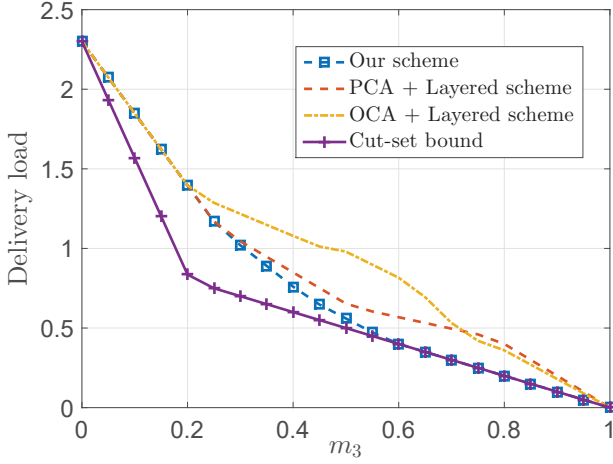*Delivery phase: We have two multicast transmissions $X_{\{1,3\},\boldsymbol{d}}$ and $X_{\{2,3\},\boldsymbol{d}}$, such that*

Fig. 4. Comparing the delivery load $R^*_{\mathfrak{A},\mathfrak{D}}(\boldsymbol{m},\boldsymbol{r})$ obtained from (21), the delivery load achieved by the two schemes in [28], and the cut-set bound [28, Theorem 2], for $K = N = 3$, $\boldsymbol{r} = [0.5, 0.8, 1]$, and $m_k = 0.8\,m_{k+1}$.

- $X_{\{1,3\},\boldsymbol{d}} = W^{(1),\{1,3\}}_{d_1,\{3\}} \oplus W^{(1),\{1,3\}}_{d_3,\{1,2\}}$, where $v_{\{1,3\}} = a^{(1)}_{\{3\}} = a^{(1)}_{\{1,2\}} = 0.1$.
- $X_{\{2,3\},\boldsymbol{d}} = W^{(1),\{2,3\}}_{d_2,\{3\}} \oplus W^{(2),\{2,3\}}_{d_2,\{2\}}$, where $v_{\{2,3\}} = a^{(1)}_{\{3\}} = a^{(2)}_{\{2\}} = 0.1$.

The optimal solution of (21) achieves the delivery load $R^*_{\mathfrak{A},\mathfrak{D}}(\boldsymbol{m},\boldsymbol{r}) = 0.2$ compared to 0.2167 which is achieved by exploiting only the intra-layer multicast opportunities. ∎

VII. CONCLUSION

In this paper, we have studied coded caching systems with heterogeneous distortion requirements. In addition to designing the caching scheme, the server allocates the sizes of the cache memories at the end users, subject to a total cache budget constraint over the set of all users. Assuming uncoded placement and linear delivery policies, we have shown that the problem of minimizing the worst-case delivery load can be formulated as a linear program.

The optimal memory allocation has been shown to follow a threshold policy. Furthermore, we have observed that our solution matches the cut-set bound in the large total cache memory budget region. As a byproduct, we have proposed a novel caching scheme which, for fixed cache sizes, outperforms the state-of-art schemes [28] by exploiting the inter-layer multicast opportunities and jointly designing the cache contents and the partitioning of the caches over the layers.

APPENDIX A
PROOF OF THEOREM 1

In order to prove that the optimal solution of (20) achieves the delivery load in Theorem 1, we first show the optimality of uniform cache allocation in each layer, i.e., $m^{(l)}_k = m^{(l)}, \forall k \in \{l, \ldots, K\}$. Then, we show that the optimal memory allocation over the layers follows the threshold policy defined by (24)-(26).

Next lemma shows the optimality of allocating equal cache sizes in a caching system where the users request the same number of bits from the desired files [2], which is equivalent to equal distortion requirements at the users [28].

Lemma 1. *For a caching system with K users, $N \geq K$ files, and cache memory budget $m_{tot} \in [0, K]$, the minimum worst-case delivery load under uncoded placement*

$$R^*_{\mathfrak{A}}(m_{tot}) = \max_{j \in [K]} \left\{ \frac{2K - j + 1}{j + 1} - \frac{(K+1)m_{tot}}{j(j+1)} \right\}, \quad (32)$$

which is achieved by uniform memory allocation and the MaddahAli-Niesen caching scheme [2].

Proof. For any memory allocation $\boldsymbol{m} = [m_1, \ldots, m_K]$,

$$R^*_{\mathfrak{A}}(\boldsymbol{m}) \geq \max_{\lambda_0 \in \mathbb{R}, \lambda_k \geq 0} \ -\lambda_0 - \sum_{k=1}^{K} m_k \lambda_k \quad (33a)$$

$$\text{subject to } \lambda_0 + \sum_{k \in \mathcal{S}} \lambda_k + \frac{K - |\mathcal{S}|}{|\mathcal{S}| + 1} \geq 0, \ \forall \mathcal{S} \subset [K], \quad (33b)$$

which is obtained by considering the average cut in the lower bound under uncoded placement in [22, Theorem 1]. In turn, by considering $\lambda_k = \lambda, \forall k \in [K]$, we get

$$R^*_{\mathfrak{A}}(m_{tot}) \geq \max_{\lambda \geq 0} \min_{j \in \{0, \ldots, K\}} \left\{ \frac{K - j}{j + 1} - \lambda(j - m_{tot}) \right\}, \quad (34)$$

$$= \max_{j \in [K]} \left\{ \frac{2K - j + 1}{j + 1} - \frac{(K+1)m_{tot}}{j(j+1)} \right\}, \quad (35)$$

where $m_{tot} = \sum_{k=1}^{K} m_k$. □

Building on Lemma 1, given $\boldsymbol{r}$ and any memory allocation $\boldsymbol{m}^{(1)}, \ldots, \boldsymbol{m}^{(K)}$, the minimum worst-case delivery load under uncoded placement, $R^*_{\mathfrak{A}}(\boldsymbol{m}^{(1)}, \ldots, \boldsymbol{m}^{(K)}, \boldsymbol{r})$ satisfies

$$R^*_{\mathfrak{A}}(\boldsymbol{m}^{(1)}, \ldots, \boldsymbol{m}^{(K)}, \boldsymbol{r}) \geq \sum_{l=1}^{K} (r_l - r_{l-1})$$

$$\max_{j \in [K-l+1]} \left\{ \frac{2(K-l+1) - j + 1}{j + 1} - \frac{(K-l+2)\sum_{k=l}^{K} m^{(l)}_k}{j(j+1)} \right\}. \quad (36)$$

Furthermore, this lower bound is achievable if we consider $m^{(l)}_k = m^{(l)}, \forall k \in \{l, \ldots, K\}$ and apply the MaddahAli-Niesen caching scheme [2] on each layer. In turn, the optimization problem in (20) simplifies to the problem of allocating the memory over the layers, which is defined as

$$\min_{t_1, \ldots, t_K} \quad \sum_{l=1}^{K} \chi_l f_l \quad (37a)$$

$$\text{subject to} \quad \sum_{l=1}^{K} t_l f_l \leq m_{tot}, \quad (37b)$$

$$0 \leq t_l \leq K - l + 1, \quad (37c)$$

where $m^{(l)} = t_l f_l/(K-l+1)$ and

$$\chi_l \triangleq \max_{j \in [K-l+1]} \left\{ \frac{2(K-l+1)-j+1}{j+1} - \frac{(K-l+2)t_l}{j(j+1)} \right\}. \quad (38)$$

Finally, we can show that the optimal solution to (37) achieves the delivery load in Theorem 1, by solving the dual of the linear program in (37).

REFERENCES

[1] Cisco, *Cisco VNI Forecast and Methodology, 2015-2020*, Jun. 2016.
[2] M. A. Maddah-Ali and U. Niesen, "Fundamental limits of caching," *IEEE Trans. Inf. Theory*, vol. 60, no. 5, pp. 2856–2867, Mar. 2014.
[3] ——, "Coding for caching: Fundamental limits and practical challenges," *IEEE Commun. Mag.*, vol. 54, no. 8, pp. 23–29, Aug. 2016.
[4] K. Wan, D. Tuninetti, and P. Piantanida, "On the optimality of uncoded cache placement," in *Proc. IEEE ITW*, 2016.
[5] Q. Yu, M. A. Maddah-Ali, and A. S. Avestimehr, "The exact rate-memory tradeoff for caching with uncoded prefetching," *arXiv:1609.07817*, 2016.
[6] ——, "Characterizing the rate-memory tradeoff in cache networks within a factor of 2," in *Proc. IEEE ISIT*, 2017.
[7] N. Karamchandani, U. Niesen, M. A. Maddah-Ali, and S. N. Diggavi, "Hierarchical coded caching," *IEEE Trans. Inf. Theory*, vol. 62, no. 6, pp. 3212–3229, Apr. 2016.
[8] M. Ji, M. F. Wong, A. M. Tulino, J. Llorca, G. Caire, M. Effros, and M. Langberg, "On the fundamental limits of caching in combination networks," in *Proc. IEEE SPAWC*, 2015.
[9] A. A. Zewail and A. Yener, "Combination networks with or without secrecy constraints: The impact of caching relays," *arXiv:1712.04930*, 2017.
[10] K. Wan, M. Ji, P. Piantanida, and D. Tuninetti, "Combination networks with caches: Novel inner and outer bounds with uncoded cache placement," *arXiv:1701.06884*, 2017.
[11] ——, "Caching in combination networks: Novel multicast message generation and delivery by leveraging the network topology," *arXiv:1710.06752*, 2017.
[12] M. Ji, G. Caire, and A. F. Molisch, "Fundamental limits of caching in wireless D2D networks," *IEEE Trans. Inf. Theory*, vol. 62, no. 2, pp. 849–869, Feb. 2016.
[13] A. A. Zewail and A. Yener, "Fundamental limits of secure device-to-device coded caching," in *Proc. IEEE Asilomar*, 2016.
[14] A. M. Ibrahim, A. A. Zewail, and A. Yener, "Device-to-device coded caching with heterogeneous cache sizes," in *Proc. IEEE ICC*, 2018.
[15] V. Ravindrakumar, P. Panda, N. Karamchandani, and V. Prabhakaran, "Fundamental limits of secretive coded caching," in *Proc. IEEE ISIT*, 2016.
[16] A. Sengupta, R. Tandon, and T. C. Clancy, "Fundamental limits of caching with secure delivery," *IEEE Trans. Inf. Forensics Security*, vol. 10, no. 2, pp. 355–370, Feb. 2015.
[17] A. A. Zewail and A. Yener, "Coded caching for resolvable networks with security requirements," in *Proc. IEEE CNS Workshops*, 2016.
[18] S. Wang, W. Li, X. Tian, and H. Liu, "Coded caching with heterogenous cache sizes," *arXiv:1504.01123*, 2015.
[19] M. M. Amiri, Q. Yang, and D. Gündüz, "Decentralized coded caching with distinct cache capacities," *arXiv:1611.01579*, 2016.
[20] A. Sengupta, R. Tandon, and T. C. Clancy, "Layered caching for heterogeneous storage," in *Proc. IEEE Asilomar*, 2016.
[21] A. M. Ibrahim, A. A. Zewail, and A. Yener, "Centralized coded caching with heterogeneous cache sizes," in *Proc. IEEE WCNC*, 2017.
[22] ——, "Coded caching for heterogeneous systems: An optimization prospective," *in preparation*.
[23] S. S. Bidokhti, M. Wigger, and A. Yener, "Gaussian broadcast channels with receiver cache assignment," *Proc. IEEE ICC*, 2017.
[24] ——, "Benefits of cache assignment on degraded broadcast channels," in *Proc. IEEE ISIT*, 2017.
[25] ——, "Benefits of cache assignment on degraded broadcast channels," *arXiv:1702.08044*, 2017.
[26] A. M. Ibrahim, A. A. Zewail, and A. Yener, "Optimization of heterogeneous caching systems with rate limited links," in *Proc. IEEE ICC*, 2017.
[27] P. Hassanzadeh, E. Erkip, J. Llorca, and A. Tulino, "Distortion-memory tradeoffs in cache-aided wireless video delivery," in *Proc. IEEE Allerton*, 2015.
[28] Q. Yang and D. Gündüz, "Coded caching and content delivery with heterogeneous distortion requirements," *arXiv:1608.05660*, 2016.
[29] R. Timo, S. S. Bidokhti, M. Wigger, and B. C. Geiger, "A rate-distortion approach to caching," *to appear in IEEE Trans. Inf. Theory*, 2018.
[30] T. M. Cover and J. A. Thomas, *Elements of information theory*. John Wiley & Sons, 2006.
[31] W. H. Equitz and T. M. Cover, "Successive refinement of information," *IEEE Trans. Inf. Theory*, vol. 37, no. 2, pp. 269–275, 1991.