Optimal Coded Caching under Statistical QoS Information

Emanuele Parrinello, Ayşe Ünsal and Petros Elia
Communication Systems Department, EURECOM, Sophia Antipolis, France
Email: {parrinel,unsal,elia}@eurecom.fr

***Abstract*— The work studies the K-user shared-link broadcast channel with coded caching, where each user's file-request comes with a certain *Quality-of-Service* (QoS) requirement, thus allowing – in the context of multi-layered coding – users to download only those file layers that are necessary to meet their own QoS requirements. The work characterizes the exact optimal worst-case delivery time, under the assumption of uncoded cache placement that is oblivious to the individual QoS requirement of each user. The work derives a new index coding based information theoretic converse, which interestingly tells us exactly how to optimally cache.**

I. Introduction

Our work is in the context of coded caching (CC) which was invented in [1] by Maddah-Ali and Niesen who considered the error-free shared-link broadcast channel (BC) with K cache-enabled users that request files from a library with N files. This work in [1] showed that pre-storing in the users' caches a fraction γ of the library, allows for simultaneously serving $K\gamma + 1$ users at a time. This was achieved with a *cache placement phase* that is based on clique covering techniques and a *delivery phase* during which each user requests a single file, and during which the server aims to deliver these files, over the bottleneck link, with a minimum possible delay. It was later shown by [2], [3] that the worst-case delivery time achieved in [1] is optimal under the constraint of uncoded cache placement. Coded caching has since then been explored in a variety of settings (see for example [4], [5], [6], [7], [8], [9], [10]).

A. Coded caching, heterogeneous distortion requirements and multi-layered coding

One interesting setting that motivates our work here is the setting where files can be downloaded at different quality levels. This is a particularly pertinent setting because most streaming services employ adaptive streaming techniques to serve videos with different quality levels, often as a function of the available bandwidth, processing power, etc. Indeed in current video coding standards such us H.264/AVC, this adaptability is a main ingredient, and the different quality levels are obtained via scalable video coding which encodes videos into many streams/layers such that the more streams the user decodes, the higher is the video quality.

This work was supported by the European Research Council under the EU Horizon 2020 research and innovation program / ERC grant agreement no. 725929.

This direction of coded caching with heterogeneous quality-of-service (distortion) requirements, was first studied in [11] by Yang and Gündüz, who presented — for the case where the users' *QoS* requirements are known during the cache-placement phase — two coded caching schemes and a cut-set type converse. Subsequent similar work in [12] proposes — for the case of the cache-aided Gaussian broadcast channel — different schemes that exploit multi-layer source coding to serve at higher rates those users with better channels. Another interesting work that merges the benefits of CC and multi-layered source coding is presented in [13] by Bayat et al. for the setting of 2-layered files, where different transmitting nodes serve K cache-enabled users who can decode the base layer and, if their channel strength allows, the enhancement layer as well. Other works that jointly study CC and multi-layered coded files can also be found in [14], [15].

Our work also considers files encoded using a layered coding technique, where each file is encoded into L layers, with the first layer providing the video stream of base-quality, and where each subsequent layer can successively refine this quality[1].

a) Caching oblivious to specific QoS requirements: Unlike other related works though, our work focuses on the case where, at the time of cache placement, the server knows only the QoS statistics (also referred to here as the *QoS profile*), in the sense that the server only knows how many users want a certain *QoS* level, but it does not know which *QoS* level each particular user wants. Having caching that is oblivious to the specific *QoS* requirement of each user, can be of particular interest because the cache placement can indeed take place long before the allowed or desired *QoS* is established.

II. System Model and Problem Definition

In particular, we consider the K-user cache-aided shared-link BC, where a transmitter with access to a library of N files $W_1, W_2, \dots, W_N$, is tasked with serving a set $\mathcal{U} = \{1, 2, \dots, K\}$ of K cache-aided users, where each user simultaneously requests a potentially different library file, at a potentially different *QoS* level. Using successive refinement source coding, each library file $W_i,\ i \in \{1, 2, \dots, N\} \triangleq [N]$

[1]The layers follow a hierarchy so that the $l+1-$th layer can only be applied if the $l-$th layer was already applied.

is separated into L component layers as follows $W_i = \{W_{i,1}, W_{i,2}, \dots, W_{i,L}\}$, where $W_{i,l},\ l \in [L]$ denotes the $l-$th layer of the $i-$th file. When a user k requests a file $W_{d_k}, d_k \in [N]$ at a *QoS* level $l \in [L]$, this will mean that this user must be succesfully delivered the first l layers $\{W_{d_k,1}, W_{d_k,2}, \dots, W_{d_k,l}\}$ of its desired file. Assuming that each file has normalized unit size $|W_i| = 1$, we will use $r_l = \sum_{j=1}^{l} |W_{i,j}|$ to denote the size of these desired l layers. Naturally $\sum_{l=1}^{L} |W_{i,l}| = r_L = |W_i| = 1,\ \forall i \in [N]$.

The *user-QoS association* is defined by the partition

$$\mathcal{U} = \{\mathcal{U}_1, \mathcal{U}_2, .., \mathcal{U}_L\} \tag{1}$$

telling us exactly the *QoS* level of each user, where users in set $\mathcal{U}_l$ must be delivered exactly (and only) layers $1, 2, \dots, l$ of their respective requested file. Related to this, the number $K_l = |\mathcal{U}_l|$ will denote the number of users with *QoS* level $l \in [L]$, and will define the *QoS profile*

$$\mathcal{L} = \{K_1, K_2, \dots, K_L\}.$$

Each *QoS profile* $\mathcal{L}$ defines a class $\mathcal{U}_\mathcal{L}$ comprising of all $\mathcal{U}$ that share the same profile[2] $\mathcal{L}$.

a) Key assumptions and objective: We will assume that each user has an isolated cache of size M (in units of 'file'), corresponding to a fraction $\gamma \triangleq M/N$ of the entire library. As it is commonly the case, we will assume that the cache placement phase is oblivious to the subsequent demand vector $\boldsymbol{d} = (d_1, d_2, \dots, d_K)$. In addition, here, we will assume that the placement is also oblivious to the user-*QoS* association $\{\mathcal{U}_1, \mathcal{U}_2, .., \mathcal{U}_L\}$ but that it knows the *QoS* profile $\mathcal{L}$. The subsequent delivery phase will commence by notifying the server of the demand vector $\boldsymbol{d}$ and of the exact user-*QoS* association $\{\mathcal{U}_1, \mathcal{U}_2, .., \mathcal{U}_L\}$.

Under these assumptions, our objective is to characterize the optimal (optimized over any caching-and-delivery strategy χ) worst-case delivery time

$$T^*(\mathcal{L}) \triangleq \min_{\chi} \max_{(\mathcal{U},\boldsymbol{d}) \in (\mathcal{U}_\mathcal{L}, \{1,\dots,N\}^K)} T(\mathcal{U}, \boldsymbol{d}, \chi) \tag{2}$$

for any given *QoS* profile $\mathcal{L}$.

III. Main Results

We proceed with the main result that identifies the optimal performance $T^*(\mathcal{L})$. This will be achieved by the scheme of Section IV and it will be capped by the matching converse of Section V.

Theorem 1. *In the K-user cache-aided BC with L-layer file coding and statistical QoS information during caching, the optimal worst-case delivery time for any profile $\mathcal{L}$, takes the form*

$$T^*(\mathcal{L}) = \sum_{l=1}^{L} \sum_{g=0}^{K} \underbrace{\frac{\sum_{r=1}^{q} \binom{K-r}{g}}{\binom{K}{g} N}}_{c_{g,l}} x^*_{g,l} \tag{3}$$

[2]We can easily see that the number of different partitions $\mathcal{U}$ associated to any fixed $\mathcal{L}$, is given by the well known multinomial coefficient $\binom{K}{K_1, \cdots, K_L}$.

where $q = min\{K - g, K - \sum_{j=1}^{l-1} K_j\}$ and where the set $\{x^*_{g,l}\}_{l \in [L], g \in \{0 \cup [K]\}}$ is the optimal point of the linear program

$$\underset{x_{g,l}}{\textit{minimize}} \quad \sum_{l=1}^{L} \sum_{g=0}^{K} c_{g,l} x_{g,l} \tag{4}$$

$$\textit{subject to} \quad \sum_{g=0}^{K} x_{g,l} = (r_l - r_{l-1})N, \quad l = 1, \dots, L \tag{5}$$

$$\sum_{g=0}^{K} g \cdot \left(\sum_{l=1}^{L} x_{g,l} \right) \leq KM, \tag{6}$$

$$x_{g,l} \geq 0, \quad l = 1, \dots, L \quad g = 0, 1, \dots, K. \tag{7}$$

where $\{x_{g,l}\}_{l \in [L], g \in \{0 \cup [K]\}}$ are the optimization variables.

The converse — as we will see in Section V — coincides with the optimal value of the linear program (LP) in (4)-(7) which, as it will become evident from Section IV, is directly used to design the scheme that optimally reflects the *QoS* statistics $\mathcal{L}$.

We complete this section by providing an achievable performance for the case where caching is oblivious to $\mathcal{L}$.

Proposition 1. *In the K-user cache-aided BC with L-layer file coding, and with caching that is oblivious to statistical QoS information, the following is achievable*

$$T_{obl}(\mathcal{L}) = \sum_{l=1}^{L} \frac{\sum_{r=1}^{q} \binom{K-r}{K\gamma}}{\binom{K}{K\gamma}} (r_l - r_{l-1}) \tag{8}$$

where q is now fixed at $q = min\{K - K\gamma, K - \sum_{j=1}^{l-1} K_j\}$.

The achievable scheme corresponding to the above proposition, is presented in Section IV as a by-product of the main scheme, essentially by properly choosing feasible values $x_{g,l}$ that do not depend on the *QoS profile*. This 'oblivious' scheme designed here, is compared in the plot in Fig. 1, to the optimal scheme that employs statistics.

IV. Optimal Scheme

In this section we present the caching and delivery schemes that achieve the optimal delay of Theorem 1. The key idea is to use the LP obtained from the information-theoretic converse in Section V (as this LP is described in (4)-(7)), as the main building block for the cache placement algorithm. The delivery algorithm builds on the aforementioned placement and modifies the delivery in [1] to serve the users at different QoS levels. An illustrative example of the cache placement and delivery scheme described in this section can be found in the extended version [16] of this work.

A. Cache Placement

The first step is to evaluate the optimal solution

$$\mathbf{x}^* \triangleq [x^*_{0,1}, x^*_{1,1}, ..., x^*_{K,1}, x^*_{0,2}, ..., x^*_{K,2}, ..., x^*_{0,L}, ..., x^*_{K,L}]$$

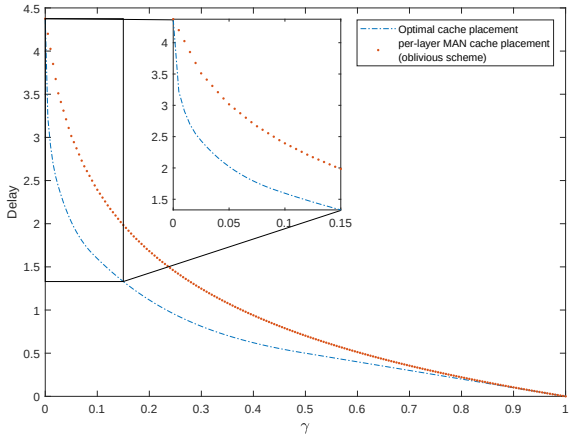

Fig. 1. Comparison of delay between the optimal and the 'oblivious' case corresponding to Proposition 1. This is done for *QoS* profile $\mathcal{L} = \{12, 10, 8, 5, 3, 2, 1\}$ and sizes $r_l = 2^{l-L}, \forall l \in [L], L = 7$.

of the LP[3]. With $\mathbf{x}^*$ in place, for any given $l \in [L]$, we split each subfile $W_{i,l}$ into $K+1$ mini-files $W_{i,l} = \{W_{i,l,g}\}_{g=0}^{K}$ such that $|W_{i,l,g}| = y_{g,l} \triangleq \frac{x^*_{g,l}}{N}$ where, due to (5), we have that $\sum_{g=0}^{K} y_{g,l} = |W_{i,l}| = r_l - r_{l-1}$. Intuitively, for each layer $l \in [L]$, we have $K+1$ redundancy levels indexed by $g \in \{0, 1, \ldots, K\}$.

Next, we further split each mini-file $W_{i,l,g}$ into $\binom{K}{g}$ equally-sized micro-files $W^{\mathcal{T}}_{i,l,g}$ as follows $W_{i,l,g} \to \{W^{\mathcal{T}}_{i,l,g} |\ |\mathcal{T}| = g, \mathcal{T} \subset [K]\}$.

At this point we can fill the cache Z_k of each user k, by placing in it, from each file W_i, all the micro-files $W^{\mathcal{T}}_{i,l,g}$ for any $\mathcal{T} \ni k$, as described below

$$Z_k \leftarrow \{W^{\mathcal{T}}_{i,l,g} \mid k \in \mathcal{T}, i \in [N], l \in [L], g \in \{0, 1, ..., K\}\}.$$

We now prove that the above placement adheres to the cache-size constraint.

1) Compliance of the placement scheme with the cache constraint: We first recall that for each mini-file $W_{i,l,g}, i \in [N], l \in [L], g \in \{0, 1, ..., K\}$, any derived micro-file $W^{\mathcal{T}}_{i,l,g}$ with $|\mathcal{T}| = g, \mathcal{T} \subset [K]$, must have size $|W^{\mathcal{T}}_{i,l,g}| = \frac{y_{g,l}}{\binom{K}{g}}$. Due to the symmetry in the cache placement, each user $k \in [K]$ stores exactly $\binom{K-1}{g-1}$ micro-files $W^{\mathcal{T}}_{i,l,g}$ from any fixed mini-file $W_{i,l,g}$, which means that, for each mini-file $W_{i,l,g}$, each user stores $\binom{K-1}{g-1}\frac{y_{g,l}}{\binom{K}{g}} = \frac{g}{K} y_{g,l}$ units of file. Consequently the total amount of data from the $l-$th layer of file $W_i, i \in [N]$ stored by each user k, takes the form $S_k(W_{i,l}) \triangleq \sum_{g=0}^{K} \frac{g}{K} y_{g,l}$, and thus the total memory employed by each user to cache a portion of a file $W_i, i \in [N]$, takes the form

$$S_k(W_i) = \sum_{l=1}^{L} S_k(W_{i,l}) = \sum_{l=1}^{L}\sum_{g=0}^{K} \frac{g}{K} y_{g,l} = \frac{1}{KN} \sum_{g=0}^{K} g \sum_{l=1}^{L} x^*_{g,l}.$$

[3]The LP in (4)-(7) has complexity linear with K ($L \cdot (K+1)$ variables to optimize).

Finally, summing over all library files, we see that each user caches

$$\sum_{i=1}^{N} \frac{1}{KN} \sum_{g=0}^{K} g \sum_{l=1}^{L} x^*_{g,l} = \frac{1}{K} \sum_{g=0}^{K} g \sum_{l=1}^{L} x^*_{g,l} \text{ (units of file)}$$

which does not exceed M due to the constraint in (6). □

Remark 1. *In the context of Proposition 1, in the absence of knowledge of $\mathcal{L}$, the cache placement is obtained from the above by substituting $\mathbf{x}^*_l$ with a vector[4] $\mathbf{x}_l$ whose only non-zero value is $N(r_l - r_{l-1})$ and is located in position $g = K\gamma$, corresponding to $x_{g,l} = 0, \forall g \in [K] \setminus \{K\gamma\}$ and $x_{K\gamma,l} = N(r_l - r_{l-1})$.*

B. Delivery Scheme

The delivery scheme sequentially serves all the requested files, and does so layer by layer, in accordance to the now known user-*QoS* association $\{\mathcal{U}_1, \mathcal{U}_2, .., \mathcal{U}_L\}$. The scheme operates in L rounds, one for each layer, where in particular, round $l \in [L]$ serves the $l-$th layer of the files requested by the users in the set $\Sigma_l \triangleq \bigcup_{j=l}^{L} \mathcal{U}_j$, which have a *QoS* level of l or above. Specifically, round l aims to deliver subfiles $W_{d_k,l}, \forall k \in \Sigma_l$.

Each round l is split into Q sub-rounds, where $Q \leq K+1$ is the total number of non-zero elements in the aforementioned $\mathbf{x}^*_l$. Consequently each sub-round g serves subfiles $W_{d_k,l,g}$ for all $k \in \bigcup_{j=l}^{L} \mathcal{U}_j$.

In each such sub-round g of round l, we apply a variation of the delivery scheme in [1], where we create $\binom{K}{g+1}$ sets $\mathcal{Q} \subseteq [K]$ of size $|\mathcal{Q}| = g+1$, and for each such set $\mathcal{Q}$, we pick the set of users $\chi_{\mathcal{Q}} = \mathcal{Q} \cap \Sigma_l$. If $\chi_{\mathcal{Q}} \neq \emptyset$, the server transmits

$$X_{\chi_{\mathcal{Q}}} = \bigoplus_{k \in \chi_{\mathcal{Q}}} W^{\mathcal{Q}\setminus\{k\}}_{d_k,l,g} \tag{9}$$

else if $\chi_{\mathcal{Q}} = \emptyset$, there is no transmission, and we move to the next $\mathcal{Q}$.

The decoding follows directly from [1].

a) Calculation of delay: We first recall that $|W_{d_k,l}| = r_l - r_{l-1}$ and that $|W_{d_k,l,g}| = \frac{x^*_{g,l}}{N}$. We also note that in each sub-round g of round l, the total number of transmissions is

$$G_{l,g} = \binom{K}{g+1} - \binom{\sum_{j=1}^{l-1} K_j}{g+1}$$

where the second term accounts for the number of times the set $\chi_{\mathcal{Q}}$ was empty. It is easy to see that if $\sum_{j=1}^{l-1} K_j < g+1$, we have $G_{l,g} = \sum_{r=1}^{K-g} \binom{K-r}{g}$, else we have

$$G_{l,g} = \binom{K-1}{g} + \binom{K-2}{g} + \cdots + \binom{\sum_{j=1}^{l-1} K_j}{g} = \sum_{r=1}^{K-\sum_{j=1}^{l-1} K_j} \binom{K-r}{g}$$

[4]$\mathbf{x}^*_l$ is the sub-vector of $\mathbf{x}^*$ composed of entries $x^*_{g,l}$, $\forall g \in \{0, 1, \ldots, K\}$.

which implies that

$$G_{l,g}=\sum_{r=1}^{min\{K-g,K-\sum_{j=1}^{l-1}K_j\}}\binom{K-r}{g}.$$

Each transmission has duration $\frac{x^*_{g,l}}{N\binom{K}{g}}$, and thus the total duration of a sub-round is

$$\frac{\sum_{r=1}^{min\{K-g,K-\sum_{j=1}^{l-1}K_j\}}\binom{K-r}{g}}{\binom{K}{g}N}x^*_{g,l}.$$

Summing over all sub-rounds and then over all rounds, we calculate the total delivery time to be the stated

$$T^*(\mathcal{L})=\sum_{l=1}^{L}\sum_{g=0}^{K}\frac{\sum_{r=1}^{min\{K-g,K-\sum_{j=1}^{l-1}K_j\}}\binom{K-r}{g}}{\binom{K}{g}N}x^*_{g,l} \quad (10)$$

which, as we will see next, is optimal.

V. Information Theoretic Converse

In this section we present the information theoretic converse that proves, in conjunction with the achievable performance in (10), the optimality of the delay in Theorem 1. Due to lack of space, some details are omitted and can be found in the extended version [16] of this work. The proof of converse draws from the technique in [2] (and its adaptation in [5]) that translates the coded caching problem into an equivalent index coding problem, making use of the cut-set type outer bound on the index coding capacity introduced in [17, Corollary 1].

The coded caching problem here is uniquely determined when the users' requests and *QoS* levels $(\boldsymbol{d},\mathcal{U})$ are revealed to the server. Focusing on the worst-case delay scenario with $N \geq K$, we define the set of worst-case demands

$$\mathcal{D}_{\mathcal{L}}=\{\boldsymbol{d}(\mathcal{U}):\boldsymbol{d}\in\mathcal{D}_{dif},\mathcal{U}\in\mathcal{U}_{\mathcal{L}}\}$$

associated to a given *QoS profile* $\mathcal{L}$, where $\mathcal{D}_{dif}$ is the set of all demand vectors $\boldsymbol{d}$ that are comprised of (the indices of) K different files.

a) Translation to index coding: A first step in converting the caching problem defined by $\boldsymbol{d}(\mathcal{U})$ into an equivalent index coding problem, is to split a requested layer of a requested file in the most general way. In particular, focusing on layer l, this means that we split any $W_{d_i,l},\ i\in\cup_{p=l}^{L}\mathcal{U}_p$ into 2^K disjoint subfiles $W^{\mathcal{T}}_{d_i,l},\mathcal{T}\in 2^{[K]}$, where $2^{[K]}$ is the power set of $[K]$, and where $\mathcal{T}\in[K]$ indicates the set of users in which $W^{\mathcal{T}}_{d_i,l}$ is cached. In the context of index coding, we can view each such subfile $W^{\mathcal{T}}_{d_i,l}$ as a message requested by a different user that has as side information all the content in the cache of the requesting user from the caching problem. Given this aforementioned representation of the requested files, the corresponding index coding problem is fully defined by the side information graph $\mathcal{G}_{\mathcal{U},\boldsymbol{d}}=(\mathcal{V}_{\mathcal{G}},\mathcal{E}_{\mathcal{G}})$, where $\mathcal{V}_{\mathcal{G}}$ (which has cardinality $|\mathcal{V}_{\mathcal{G}}|=2^{K-1}\cdot\sum_{j=1}^{L}jK_j$) is the set of vertices corresponding to the requested subfiles $W^{\mathcal{T}}_{d_i,l}$, and where $\mathcal{E}_{\mathcal{G}}$ is the set of directed edges of the graph. A directed edge from vertex $v\in\mathcal{V}_{\mathcal{G}}$ to $v'\in\mathcal{V}_{\mathcal{G}}$ exists if and only if the index coding user requesting the subfile corresponding to vertex v', knows the subfile corresponding to vertex v.

Before proceeding with the proof, we recall a useful lemma from [17], which we adapt to our setting.

Lemma 1. *For a given $\mathcal{U},\boldsymbol{d},\chi$, with a corresponding side information graph $\mathcal{G}_{\mathcal{U},\boldsymbol{d}}=(\mathcal{V}_{\mathcal{G}},\mathcal{E}_{\mathcal{G}})$, the inequality*

$$T\geq\sum_{\mathcal{V}\in\mathcal{V}_{\mathcal{J}}}|\mathcal{V}| \quad (11)$$

holds for every acyclic induced subgraph $\mathcal{J}$ of $\mathcal{G}_{\mathcal{U},\boldsymbol{d}}$, where $\mathcal{V}_{\mathcal{J}}$ denotes the set of nodes of the subgraph $\mathcal{J}$, and where $|\mathcal{V}|$ is the size of the message/subfile/node $\mathcal{V}$.

Lemma 1 will be used to lower bound the delay $T(\mathcal{U},\boldsymbol{d},\chi)$ associated to any pair $\mathcal{U},\boldsymbol{d}$. To this end, we proceed to carefully select acyclic subgraphs that will yield, via Lemma 1, tighter lower bounds for $T(\mathcal{U},\boldsymbol{d},\chi)$. The following lemma considers permutations $\boldsymbol{\sigma}=(\sigma_1,\sigma_2,\ldots,\sigma_K)$ on vector $(1,2,\ldots,K)$.

Lemma 2. *All subfiles*

$$W^{\mathcal{T}_i}_{d_{\sigma_i},l},\ \forall i\in[K],\ \forall l:\sigma_i\in\bigcup_{p=l}^{L}\mathcal{U}_p$$

such that

$$\sigma_i\in\mathcal{U}_p\ \wedge\ \sigma_j\in\mathcal{U}_w\ with\ i\leq j\iff p\geq w \quad (12)$$
$$and\qquad \mathcal{T}_i\subseteq[K]\setminus\{\sigma_1,\ldots,\sigma_i\}$$

compose an acyclic subgraph $\mathcal{J}$ of $\mathcal{G}$.

The proof of Lemma 2 follows directly from Lemma 1 in [2] for the construction of acyclic subgraphs according to a generic $\boldsymbol{\sigma}$, while the specific choice of permutations $\boldsymbol{\sigma}$, which adheres to the condition in (12), directly draws from [5][5].

In the above, for a given *QoS profile* $\mathcal{L}$, the total number of permutations $\boldsymbol{\sigma}$ that satisfy the condition in (12) can be easily calculated to be $K_1!K_2!\cdots K_L!$. We will denote the set of all such permutations by Σ.

After choosing an acyclic subgraph according to Lemma 2, we return to Lemma 1 and form the following lower bound

$$T(\mathcal{U},\boldsymbol{d},\chi)\geq T^{lb}_{\boldsymbol{\sigma}}(\mathcal{U},\boldsymbol{d},\chi) \quad (13)$$

where, assuming for notational simplicity that $|\mathcal{U}_L|=1$, we get that

$$T^{lb}_{\boldsymbol{\sigma}}(\mathcal{U},\boldsymbol{d},\chi)\triangleq\sum_{l=1}^{L}\sum_{\mathcal{T}_1\subseteq[K]\setminus\{\sigma_1\}}|W^{\mathcal{T}_1}_{d_{\sigma_1},l}|+\sum_{l=1}^{L-1}\sum_{\mathcal{T}_2\subseteq[K]\setminus\{\sigma_1,\sigma_2\}}|W^{\mathcal{T}_2}_{d_{\sigma_2},l}|$$
$$+\cdots+\sum_{l=1}^{1}\sum_{\mathcal{T}_K\subseteq[K]\setminus\{\sigma_1,\ldots,\sigma_K\}}|W^{\mathcal{T}_K}_{d_{\sigma_K},l}|. \quad (14)$$

Since (14) holds for each $\boldsymbol{\sigma}\in\Sigma$, we proceed to lower bound $T(\mathcal{U},\boldsymbol{d},\chi)$ with the following average

$$T(\mathcal{U},\boldsymbol{d},\chi)\geq\frac{1}{|\Sigma|}\sum_{\boldsymbol{\sigma}\in\Sigma}T^{lb}_{\boldsymbol{\sigma}}(\mathcal{U},\boldsymbol{d},\chi). \quad (15)$$

[5]What is different from [5] is that here, for each $\boldsymbol{\sigma}$, we choose to aggregate several maximal acyclic subgraphs.

We now recall that our interest lies on the worst-case delay scenario for a given profile $\mathcal{L}$. Hence, we can lower bound the optimal worst-case delay as

$$T^*(\mathcal{L}) \triangleq \min_{\chi} \max_{(\mathcal{U},\boldsymbol{d})\in(\mathcal{U}_{\mathcal{L}},[N]^K)} T(\mathcal{U},\boldsymbol{d},\chi) \tag{16}$$

$$\geq \min_{\chi} \frac{1}{|\mathcal{D}_{\mathcal{L}}|} \sum_{\boldsymbol{d}(\mathcal{U})\in\mathcal{D}_{\mathcal{L}}} T(\boldsymbol{d}(\mathcal{U}),\chi). \tag{17}$$

Since $|\mathcal{U}_{\mathcal{L}}| = \binom{K}{K_1,K_2,\ldots,K_L}$, we can easily see that the set $\mathcal{D}_{\mathcal{L}}$ has cardinality[6] $P(N,K)\cdot\binom{K}{K_1,K_2,\ldots,K_L}$, and thus the above extends to

$$T^*(\mathcal{L}) = \min_{\chi} T(\mathcal{L},\chi) \tag{18}$$

$$\geq \min_{\chi} \frac{1}{P(N,K)\binom{K}{K_1,K_2,\ldots,K_L}K_1!K_2!\ldots K_L!}\times$$
$$\times \sum_{\boldsymbol{d}(\mathcal{U})\in\mathcal{D}_{\mathcal{L}}} \sum_{\boldsymbol{\sigma}\in\Sigma} T^{lb}_{\boldsymbol{\sigma}}(\boldsymbol{d}(\mathcal{U}),\chi) \tag{19}$$

where $T^{lb}_{\boldsymbol{\sigma}}(\boldsymbol{d}(\mathcal{U}),\chi)$ is given by (14).

Due to symmetry, all the subfiles that are cached in exactly g users, will appear an equal number of times in the summation shown in (19), and thus in (19) the coefficients (after expanding (19) by applying (14)) — in front of each subfile term $|W^{\mathcal{T}}_{j,l}|$ with a fixed $|\mathcal{T}| = g,\ g\in\{0,1,\ldots,K\}$ — are identical.

We can then easily calculate that in (14), for any $l\in[L]$, there are $\sum_{r=1}^{min\{K-g,K-\sum_{j=1}^{l-1}K_j\}}\binom{K-r}{g}$ subfile terms $|W^{\mathcal{T}}_{j,l}|$ for which $|\mathcal{T}| = g$. Consequently, since there exist in total $\binom{K}{g}N$ subfiles $W^{\mathcal{T}}_{j,l}$ with $|\mathcal{T}| = g$, in the sum in (19), the coefficient of each $|W^{\mathcal{T}}_{j,l}|$ with $|\mathcal{T}| = g$ is

$$P(N,K)K!\frac{\sum_{r=1}^{min\{K-g,K-\sum_{j=1}^{l-1}K_j\}}\binom{K-r}{g}}{\binom{K}{g}N}. \tag{20}$$

For $x_{g,l} \triangleq \sum_{n\in[N]}\sum_{\mathcal{T}\subseteq[K]:|\mathcal{T}|=g}|W^{\mathcal{T}}_{n,l}|$ being the total amount of data of layer $l\in[L]$ stored in exactly g users, we can combine (14), (19), (20) to get

$$T(\mathcal{L},\chi) \geq \sum_{l=1}^{L}\sum_{g=0}^{K} \underbrace{\frac{\sum_{r=1}^{min\{K-g,K-\sum_{j=1}^{l-1}K_j\}}\binom{K-r}{g}}{\binom{K}{g}N}}_{c_{g,l}} x_{g,l}. \tag{21}$$

In terms of constraints, the successive refinement source coding applied to the files, implies the following equalities

$$\sum_{g=0}^{K} x_{g,l} = (r_l - r_{l-1})N, \quad \forall l\in\{1,\ldots,L\} \tag{22}$$

while the sum cache size constraint forces

$$\sum_{g=0}^{K} g\cdot\left(\sum_{l=1}^{L} x_{g,l}\right) \leq KM. \tag{23}$$

[6]We denote the number of $k-$permutations of n as $P(n,k) = \frac{n!}{(n-k)!}$.

Finally, by combining (18), (21) and (22), (23), the desired lower bound can be derived from the following linear program

$$\begin{aligned} \underset{x_{g,l}}{\text{minimize}} \quad & \sum_{l=1}^{L}\sum_{g=0}^{K} c_{g,l}x_{g,l} \\ \text{subject to} \quad & (22),(23), \\ & x_{g,l}\geq 0, \quad l=1,...,L. \end{aligned}$$

This concludes the proof. □

VI. Conclusions

In this work, we characterized the rate-memory tradeoff for the coded caching problem with multi-layer coded files for the case when, in the uncoded cache placement, the server knows only *QoS* statistics but does not know the *QoS* requirement of each user. To this end, we developed an information theoretic converse which in turn defined the design of an optimal scheme by defining how much of each layer should be placed in the caches. This interesting back-and-forth between the converse and the scheme, nicely highlights the usefulness of finding exact information theoretic bounds, since such exact bounds may have the potential to recreate the structure of the optimal scheme.

References

[1] M. Maddah-Ali and U. Niesen, "Fundamental limits of caching," *IEEE Trans. Inf. Theory*, May 2014.

[2] K. Wan, D. Tuninetti, and P. Piantanida, "On the optimality of uncoded cache placement," in *Inf. Theory Workshop (ITW), IEEE*, Sep 2016.

[3] Q. Yu, M. A. Maddah-Ali, and A. S. Avestimehr, "The exact rate-memory tradeoff for caching with uncoded prefetching," *IEEE Trans. Inf. Theory*, Feb 2017.

[4] S. P. Shariatpanahi, S. A. Motahari, and B. H. Khalaj, "Multi-server coded caching," *IEEE Trans. Inf. Theory*, Dec 2016.

[5] E. Parrinello, A. Ünsal, and P. Elia, "Fundamental limits of caching in heterogeneous networks with uncoded prefetching," *arXiv preprint https://arxiv.org/pdf/1811.06247.pdf*, 2018.

[6] J. Hachem, N. Karamchandani, and S. Diggavi, "Coded caching for multi-level popularity and access," *IEEE Trans. Inf. Theory*, May 2017.

[7] J. Zhang, X. Lin, and X. Wang, "Coded caching under arbitrary popularity distributions," *IEEE Trans. Inf. Theory*, Jan 2018.

[8] Q. Yan, M. Cheng, X. Tang, and Q. Chen, "On the placement delivery array design for centralized coded caching scheme," *IEEE Trans. Inf. Theory*, Sept 2017.

[9] E. Lampiris and P. Elia, "Adding transmitters dramatically boosts coded-caching gains for finite file sizes," *J. on Selec. Areas in Comm.*, 2018.

[10] A. Tolli, S. P. Shariatpanahi, J. Kaleva, and B. Khalaj, "Multicast beamformer design for coded caching," in *Proc. IEEE Int. Symp. on Inform. Theory (ISIT)*, June 2018.

[11] Q. Yang and D. Gündüz, "Coded caching and content delivery with heterogeneous distortion requirements," *IEEE Trans. Inf. Theory*, 2018.

[12] M. M. Amiri and D. Gündüz, "On the capacity region of a cache-aided gaussian broadcast channel with multi-layer messages," in *Proc. IEEE Int. Symp. on Inform. Theory (ISIT)*, June 2018.

[13] M. Bayat, C. Yapar, and G. Caire, "Spatially scalable lossy coded caching," in *Int. Symp. on Wireless Comm. Systems (ISWCS)*, Aug 2018.

[14] A. M. Ibrahim, A. A. Zewail, and A. Yener, "On coded caching with heterogeneous distortion requirements," in *2018 Inf. Theory and Appl. Workshop (ITA)*, Feb 2018.

[15] D. Cao, D. Zhang, P. Chen, N. Liu, W. Kang, and D. Gunduz, "Coded caching with heterogeneous cache sizes and link qualities: The two-user case," in *Proc. IEEE Int. Symp. on Inform. Theory (ISIT)*, June 2018.

[16] E. Parrinello, A. Ünsal, and P. Elia, "Optimal coded caching under statistical QoS information," *Manuscript in Preparation (arxiv)*, 2019.

[17] F. Arbabjolfaei, B. Bandemer, Y. H. Kim, E. Şaşoğlu, and L. Wang, "On the capacity region for index coding," in *Proc. IEEE Int. Symp. on Inform. Theory (ISIT)*, Jul 2013.