Coded Caching for Files with Distinct File Sizes

Jinbei Zhang†, Xiaojun Lin‡, Chih-Chun Wang‡, Xinbing Wang†
†Department of Electronic Engineering, Shanghai Jiao Tong University, China
‡School of Electrical and Computer Engineering, Purdue University, USA
Email: †{abelchina, xwang8}@sjtu.edu.cn, ‡{linx,chihw}@purdue.edu

***Abstract*—Coded caching can exploit new multicast opportunities even when multiple users request different pieces of content, and thus can significantly reduce the backhaul requirement for serving high-volume content. However, existing studies of coded caching have been limited to the scenarios where all files of interest are of a common size. This work studies the performance limits of coded caching when the file sizes are different. We derive a new lower bound and an achievable upper bound for the worst-case transmission rate under coded caching, and show that these two bounds differ by at most a $\Theta(\log K)$ factor, where K is the number of users in the system. There are two key novelties in our analysis. First, our lower bound is derived by considering a new cut-set bound where larger files are requested more times. The analysis of this new cut-set bound requires careful concatenation of several entropy inequalities. Compared to a lower bound using standard cut-set arguments, our lower bound is improved by a $\Theta(\log K)$ factor. Second, our achievable scheme uses a caching probability that increases proportionally with the file size. Compared to schemes that use a common caching probability, the achievable rate of our scheme is reduced by a $\Theta(\frac{K}{\log^2 K})$ factor.**

I. Introduction

A new form of caching schemes, called "coded caching", have received significant attention lately in reducing the backhaul requirement for serving a large volume of content to multiple users [1]. The significant performance improvement of coded caching arises from its capability to exploit potential multicast opportunities even when each user requests a *different* piece of content. Consider the setting of [1] where one server serves K users via a broadcast channel. Each user has a storage/cache of size M bits. The server has N files ($N > K$) with equal size F bits ($F > \frac{M}{N}$), and each user can request any one of the N files. Note that in the worst case, each user may request a distinct file. In conventional uncoded caching schemes, the server would be unable to exploit the broadcast capability of the channel in such a worst case, and thus had to transmit to each user a difference piece of content that is not stored in the user's cache. It is then easy to see that the total transmission rate needed from the server in the worst case is $KF(1 - \frac{M}{NF})$, since each user can only cache $\frac{M}{NF}$ fraction of each file. In contrast, in a coded caching scheme [1], the worst-case transmission rate from the server is reduced to $KF \cdot \frac{1-M/NF}{1+KM/NF}$. The additional reduction factor of $1/[1 + KM/(NF)]$ is significant when the global storage capability of the system is large, and thus is called the *global caching gain* in [1]. The key idea behind this performance improvement is to exploit "multicast" opportunities even when different users request different files. For example, suppose that there are two users A, B, and each user requests a different file. If user A has cached some content requested by user B, and user B has cached some content requested by user A, the server can broadcast the XOR of these two parts, which allows both users to decode their requested content.

Along this line, [1]–[7] have studied the fundamental limits of coded caching under the assumption that all files are of the same size. A common theme of these studies is to first find an information-theoretic lower bound of the transmission rate. Then they characterize the performance of an achievable scheme with a common caching probability, i.e., which caches a common fraction of every content in each user's cache. Finally they prove that the performance of the achievable scheme is a constant factor away from the lower bound. Such approaches are later generalized to decentralized coded caching schemes [2], hierarchical networks [3], and multi-level coded caching [4], average rate [5], [6] and online caching [7], respectively.

However, one limitation of these previous studies is that they assume all files to be of the same size. In practice, it is common that different types of content are of significantly different sizes. In this paper, we carry out the first study of the performance limits of coded caching when the files are of different sizes. We derive a lower bound and an achievable upper bound for the worst-case transmission rate that differ by at most a $\Theta(\log K)$ factor. The contributions of our results are two-fold. First, our lower bound (Proposition 1) considers a new cut-set bound that is different from those used in prior work. As a result, we tighten the lower bound by a $\Theta(\log K)$ factor (see the comparison with Lemma 1). In contrast to the cut-set bounds in [1]–[7] where each file is requested only once, in our new cut-set bound each file could be requested multiple times, in proportion to its file size. Analyzing this cut-set bound also requires careful concatenation of several entropy inequalities [8] and could be of independent interest. Second, our achievable upper bound (see Sec. IV) uses a caching probability that is proportional to the file size. In other words, the amount of cached content for a file is *quadratically* proportional to its file size. In contrast to a scheme that uses a common caching probability for all files, the transmission rate of our achievable scheme is reduced by a $\Theta(K/\log^2 K)$ factor.

As illustrated in Table I, for a system with 8 users and 2 types of file sizes (the detailed setting provided in Section V), our new lower bound (LB2, Proposition 1) is higher than the conventional lower bound (LB1, Lemma 1) by about 11%, and

our achievable scheme (UB2) attains a transmission rate lower than that using a common caching probability (UB1) by about 16%. For a system with 8 users and 3 types of file sizes, our lower bound is 30% higher, and our achievable upper bound is about 31% lower. The trend when the number of users and number of file types further increase is illustrated in Figure 1.

These results thus provide significant new insights in the performance limits of coded caching in the more practical scenarios of distinct file sizes. The rest of the paper is organized as follows. In Section II, we present the network model. We provide the lower bounds of the worst-case transmission rate in Section III, and derive the achievable rate in Section IV. Numerical comparison is presented in Section V. Finally, we conclude.

TABLE I: Comparisons

Rate	LB2/LB1	Gain	UB1/UB2	Gain
2 types	4/3.6	11%	15.3757/13.2523	16%
3 types	5/3.8462	30%	21.2593/16.2196	31%

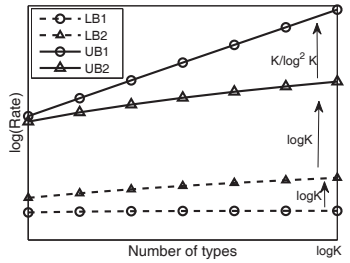

Fig. 1: An illustration on the trend when the number of types increases.

II. Network Model

We assume there are N files from the set $\mathbb{F} = \{\mathcal{F}_1, \mathcal{F}_2, ..., \mathcal{F}_N\}$. The size of the i-th file $\mathcal{F}_i$ is denoted by $F_i \triangleq |\mathcal{F}_i|$. Without loss of generality, we assume that the file size is non-increasing, i.e., $F_i \geq F_j$ if $i \leq j$.

All the files are stored in a server, which serves K users through a broadcast channel. Each user is equipped with a cache of a common size M. The content of the data cached in user k is denoted by $\mathcal{M}_k$. We assume that $\sum_{i=1}^{N} F_i \geq 2M$ and $N \geq K$, which is true in most situations where cache size is limited. During off-peak hours, the server can place some of the contents (possibly coded) in each user's cache $\mathcal{M}_k$, with the hope of reducing the rate needed to satisfy users' requests during peak hours. This process is called the caching phase. We emphasize that the caching phase must be completed before any file requests are made.

During peak hours, the k-th user will request the content of the w_k-th file, namely $\mathcal{F}_{w_k}$. Denote the requests of all K users as a K-dimensional vector $\vec{w}$. Since there are N files to choose from, there are totally N^K request patterns $\vec{w}$, and we denote the collection of all patterns by $\mathbb{W}$. If part of the file $\mathcal{F}_{w_k}$ has been cached in $\mathcal{M}_k$, it will be retrieved locally. For those uncached content, the server will broadcast some additional content, denoted by $\mathcal{R}$, to all K users simultaneously. The goal is that every user k must be able to reconstruct the file $\mathcal{F}_{w_k}$, based on the content $\mathcal{R}$ received during peak hours and the cached data $\mathcal{M}_k$. Obviously, what content $\mathcal{R}$ to transmit depends on the request pattern $\vec{w}$ and on the set of files $\mathbb{F}$. As a result, we often denote it by $\mathcal{R}_{\vec{w}}(\mathbb{F})$.

From the above discussion, given a set of files $\mathbb{F}$, a coded caching scheme needs to decide (a) what is the content to be cached in each $\mathcal{M}_k, k = 1, ..., K$; and (b) for each pattern $\vec{w} \in \mathbb{W}$, what is the additional data to send, i.e., $\{\mathcal{R}_{\vec{w}}(\mathbb{F}) : \forall \vec{w} \in \mathbb{W}\}$. The objective is to minimize the worst-case transmission rate $\max_{\vec{w}\in\mathbb{W}} \mathcal{R}_{\vec{w}}(\mathbb{F})$, denoted by $R(\mathbb{F})$. We said that the rate $R(\mathbb{F})$ is achievable if, for every $\varepsilon > 0$ and every large enough file size, there exists a caching and transmission scheme such that, regardless of the request pattern $\vec{w}$, with probability of error less than ε, every user can reconstruct the requested file. Let $R^*(\mathbb{F})$ denote the infimum over all achievable $R(\mathbb{F})$.

A. Two Approximate Systems

In the previous formulation, the minimal rate $R^*(\mathbb{F})$ depends on the sizes of the N files and we did not impose any constraints on the file sizes. We now consider two quantized versions from $\mathbb{F}$, constructed as

$$\begin{aligned} \mathbb{F}^{\text{UB}} &= \{\mathcal{F}_i^{\text{UB}} | F_i^{\text{UB}} = F_1 \cdot 2^{-\lfloor \log_2 \frac{F_1}{F_i} \rfloor}, 1 \leq i \leq N\}, \\ \mathbb{F}^{\text{LB}} &= \{\mathcal{F}_i^{\text{LB}} | F_i^{\text{LB}} = F_1 \cdot 2^{-\lfloor \log_2 \frac{F_1}{F_i} \rfloor - 1}, 1 \leq i \leq N\}. \end{aligned} \tag{1}$$

It is easy to see that $F_i^{\text{LB}} \leq F_i \leq F_i^{\text{UB}}$, for any $i \leq N$. This thus naturally implies $R^*(\mathbb{F}^{\text{LB}}) \leq R^*(\mathbb{F}) \leq R^*(\mathbb{F}^{\text{UB}})$.

Under certain additional conditions, one can show that the two rates $R^*(\mathbb{F}^{\text{LB}})$ and $R^*(\mathbb{F}^{\text{UB}})$ differ by at most a constant term since $F_i^{\text{LB}} = F_i^{\text{UB}}/2$ for all i. Hence, below we will mainly focus on those file sets $\mathbb{F}$ such that the file sizes differ by power-of-2 factors.

Thus, in the sequel we will only consider those $\mathbb{F}$ with the above property. Assume that there are at most T distinct file sizes in $\mathbb{F}$. We now call the files of the same size as "of the same type". Thus, we introduce the following equivalent notations. The l-th type, $l = 1, ..., T$, has file size $F_l = 2^{1-l} F_1$ where F_1 is the largest file size in the system. Suppose that the l-th type has N_l different files. Then, the collection of all N_l files of type l is denoted by $\mathbb{F}_l$ and we re-index each file of the l-th type by $\mathbb{F}_l = \{\mathcal{F}_{lj} | 1 \leq j \leq N_l\}$.

III. Lower Bounds Of the Worst-Case Transmission Rate

We now present two lower bounds on $R^*(\mathbb{F})$. The first lower bound (Lemma 1) is derived using standard cut-set bounds in the literature, which is then compared with the second and tighter lower bound (Prop. 1). We then highlight the difference in the analysis.

Lemma 1: The worst-case transmission rate is lower bounded by

$$R^*(\mathbb{F}) \geq \max_{\Phi} \frac{1}{\left\lceil \sum_{l\in\Phi} N_l/L \right\rceil} \left(\sum_{l\in\Phi} N_l F_l - LM \right) \tag{2}$$

where $L = \left\lfloor \sum_{l\in\Phi} N_l F_l / 2M \right\rfloor$, and the maximization is taken over any subset Φ of $\{1, 2, ..., T\}$ that satisfies $2M \leq \sum_{l\in\Phi} N_l F_l \leq 2KM$. If the expressions inside the floor and ceil functions in (2) are integers, the lower bound in (2) can be represented as

$$R^*(\mathbb{F}) \geq \max_{\Phi} \frac{(\sum_{l\in\Phi} N_l F_l)^2}{4M \sum_{l\in\Phi} N_l}. \tag{3}$$

The lower bound in Lemma 1 can be easily obtained using standard cut-set bounds in the literature, which we provide a high-level sketch below. For a subset Φ of file types, we choose h ($h \leq K$) users to request files from $\mathbb{F}_l$, $l \in \Phi$. As in prior studies [1]–[7], we can construct $\left\lceil \frac{\sum_{l\in\Phi} N_l}{h} \right\rceil$ request patterns for these h users such that every file in $\mathbb{F}_l$, $l \in \Phi$, is requested once. Since all the files requested can be retrieved from the transmissions and the local storages, we have the following cut-set, i.e., $\left\lceil \frac{\sum_{l\in\Phi} N_l}{h} \right\rceil R^*(\mathbb{F}) + hM \geq \sum_{l\in\Phi} N_l F_l$. Choosing $h = L$, we obtain (2).

We next present the second and tighter lower bound. For ease of exposition, we first focus on the case when the following two assumptions hold.

Assumption 1: $\frac{2M}{F_1}$ and $\frac{N_l F_l}{2M}$ are both integers for all types l such that $1 \leq l \leq \min(T, \log_2 K)$.

Assumption 2: $\sum_{l=1}^{\min(T,\log_2 K)} \frac{N_l F_l}{2M} \leq K$.

With **Assumption 1**, we will not need to be concerned with the use of floor and ceiling functions, which simplifies the discussions below. On the other hand, **Assumption 2** is similar to the constraint on the set Φ in Lemma 1. While **Assumptions 1-2** simplify the analysis of Proposition 1, they still allow us to expose the key new insights of the proof. We will then relax **Assumptions 1-2** in Proposition 2.

Proposition 1: Under **Assumptions 1-2**, the worst-case transmission rate can be lower bounded by

$$R^*(\mathbb{F}) \geq \sum_{l=1}^{\min(T,\log_2 K)} \frac{N_l F_l^2}{4M}. \tag{4}$$

To compare (3) and (4), we let $T = \log_2 K$ and $N_{l+1} = 4N_l$. Recall that $F_{l+1} = \frac{1}{2} F_l$. Then, the file set Φ chosen in Lemma 1 equals to $\{1, 2, ..., \log_2 K\}$. The RHS of (3) is smaller than $\frac{N_1 F_1^2}{M}$. On the other hand, the RHS of (4) equals to $\log_2 K \cdot \frac{N_1 F_1^2}{4M}$, which is a $\Theta(\log_2 K)$ factor higher than (3).

The key novelty in the proof of Proposition 1 is to use a new cut-set bound that is different from that in (2). Unlike the derivation of (2) where we consider a set of request patterns so that every file is requested once, in the new cut-set bound we consider a new set of request patterns so that larger files are requested more times. The study of this new scenario also requires more careful concatenation of several entropy inequalities [8], some of which have been used in [1] and [4]. For ease of exposition, next we focus on the simpler case with only two types $T = 2$, which however still illustrates the novelty of our constructions.

Let $s_1 = \frac{N_1 F_1}{2M}$ and $s_2 = \frac{N_2 F_2}{2M}$. By **Assumption 1**, both s_1 and s_2 are integers and $\min(s_1, s_2) \geq 1$. We then choose two user sets $\mathcal{U}_1$ and $\mathcal{U}_2$, which satisfy $|\mathcal{U}_1| = s_1, |\mathcal{U}_2| = s_2, \mathcal{U}_1 \bigcap \mathcal{U}_2 = \emptyset$. By **Assumption 2**, we can always find $\mathcal{U}_1$ and $\mathcal{U}_2$ among the K different users.

Divide file set $\mathbb{F}_2$ into two non-overlapping subsets $\mathbb{F}_{21}$ and $\mathbb{F}_{22}$ with equal size $N_2/2$. We then choose two *disjoint* sets of request pattern $\mathcal{D}_1$ and $\mathcal{D}_2$, which satisfy $|\mathcal{D}_1| = |\mathcal{D}_2| = \frac{N_1}{s_1}$ and $\bigcup_{\vec{w}\in\mathcal{D}_1} \bigcup_{k\in\mathcal{U}_1} \mathcal{F}_{w_k} = \mathbb{F}_1$, $\bigcup_{\vec{w}\in\mathcal{D}_1} \bigcup_{k\in\mathcal{U}_2} \mathcal{F}_{w_k} = \mathbb{F}_{21}$, $\bigcup_{\vec{w}\in\mathcal{D}_2} \bigcup_{k\in\mathcal{U}_1} \mathcal{F}_{w_k} = \mathbb{F}_1$, $\bigcup_{\vec{w}\in\mathcal{D}_2} \bigcup_{k\in\mathcal{U}_2} \mathcal{F}_{w_k} = \mathbb{F}_{22}$. We first explain the construction of $\mathcal{D}_1$. In the very first request pattern $\vec{w}$ in $\mathcal{D}_1$, we let the s_1 users in $\mathcal{U}_1$ request the first s_1 files in $\mathbb{F}_1$, and let the s_2 users in $\mathcal{U}_2$ request the first s_2 files in $\mathbb{F}_{21}$. Then, in the second request $\vec{w}$ in $\mathcal{D}_1$, we let the s_1 users in $\mathcal{U}_1$ requests the second s_1 files in $\mathbb{F}_1$ and let the s_2 users in $\mathcal{U}_2$ request the second s_2 files in $\mathbb{F}_{21}$. Continue this construction until $|\mathcal{D}_1| = N_1/s_1$, i.e., each of the N_1 files in $\mathbb{F}_1$ has all been requested once. At the same time, totally $\frac{N_1}{s_1} s_2$ files of $\mathbb{F}_2$ have been requested by users in $\mathcal{U}_2$. Since $s_1 = \frac{N_1 F_1}{2M}$, $s_2 = \frac{N_2 F_2}{2M}$ and $F_1 = 2F_2$, we have $\frac{N_1}{s_1} s_2 = N_2/2$. That is, all files in the first half $\mathbb{F}_{21}$ have been requested.

The construction of $\mathcal{D}_2$ is similar. The difference is that, *we allow the files in $\mathbb{F}_1$ to be requested by users in $\mathcal{U}_1$ the second time during $\mathcal{D}_2$*, while the users in $\mathcal{U}_2$ will now request the second half $\mathbb{F}_{22}$ instead. See Figure 2 for illustration.

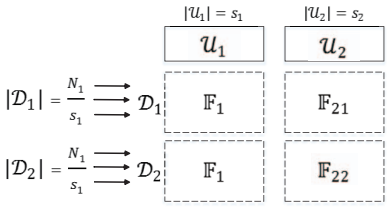

Fig. 2: An illustration on the requesting process for file systems with two types.

However, even with this new set of request patterns, new analysis is needed in order to produce a better lower bound on the transmission rate. Specifically, if we directly apply the idea that the overall rate from all the transmissions plus all the cache sizes must be greater than the total size of all files combined (as in the derivation of (2)), we would obtain

$$\frac{2N_1}{s_1} R^*(\mathbb{F}) + \frac{N_1 F_1}{2} + \frac{N_2 F_2}{2} \geq N_1 F_1 + N_2 F_2. \tag{5}$$

On the other hand, a strictly better bound than (5) can be obtained, as stated in the following lemma.

Lemma 2: For the request patterns constructed as in Figure 2, the worst-case transmission rate should satisfy

$$\frac{2N_1}{s_1} R^*(\mathbb{F}) \geq N_1 F_1 + \frac{N_2 F_2}{2}. \tag{6}$$

Thus, the lower bound on $\frac{2N_1}{s_1} R^*(\mathbb{F})$ is increased by another term $\frac{N_1 F_1}{2}$. This increase is the key step towards the tighter lower bound in Proposition 1. Indeed, substituting $s_1 = \frac{N_1 F_1}{2M}$ and noting that $F_1 = 2F_2$, the $T = 2$ case of Proposition 1 follows immediately from Lemma 2, i.e., $R^*(\mathbb{F}) \geq \frac{N_1 F_1^2 + N_2 F_2^2}{4M}$.

Proof of Lemma 2: For ease of presentation, we denote $\mathcal{R}_{\mathcal{D}_1} \triangleq \bigcup_{\vec{w}\in\mathcal{D}_1} \mathcal{R}_{\vec{w}}$, $\mathcal{R}_{\mathcal{D}_2} \triangleq \bigcup_{\vec{w}\in\mathcal{D}_2} \mathcal{R}_{\vec{w}}$, $\mathcal{M}_{\mathcal{U}_1} \triangleq \bigcup_{k\in\mathcal{U}_1} \mathcal{M}_k$, and $\mathcal{M}_{\mathcal{U}_2} \triangleq \bigcup_{k\in\mathcal{U}_2} \mathcal{M}_k$. Summing over all rates for each transmission, we have

$$\begin{aligned}
&\frac{2N_1}{s_1}R^*(\mathbb{F}) \geq H(\mathcal{R}_{\mathcal{D}_1}) + H(\mathcal{R}_{\mathcal{D}_2}) \\
&= H(\mathcal{R}_{\mathcal{D}_1}|\mathbb{F}_1) + I(\mathcal{R}_{\mathcal{D}_1};\mathbb{F}_1) + H(\mathcal{R}_{\mathcal{D}_2}|\mathbb{F}_1) + I(\mathcal{R}_{\mathcal{D}_2};\mathbb{F}_1) \\
&\geq H(\mathcal{R}_{\mathcal{D}_1}\cup\mathcal{R}_{\mathcal{D}_2}|\mathbb{F}_1) + I(\mathcal{R}_{\mathcal{D}_1};\mathbb{F}_1) + I(\mathcal{R}_{\mathcal{D}_2};\mathbb{F}_1) \\
&= I(\mathcal{R}_{\mathcal{D}_1}\cup\mathcal{R}_{\mathcal{D}_2};\mathbb{F}_2|\mathbb{F}_1) + I(\mathcal{R}_{\mathcal{D}_1};\mathbb{F}_1) + I(\mathcal{R}_{\mathcal{D}_2};\mathbb{F}_1).
\end{aligned} \tag{7}$$

Here, the first, second and third lines follow from the definition and basic properties of entropy. The fourth line is due to $H(\mathcal{R}_{\mathcal{D}_1}\cup\mathcal{R}_{\mathcal{D}_2}|\mathbb{F}_1) = H(\mathcal{R}_{\mathcal{D}_1}\cup\mathcal{R}_{\mathcal{D}_2}|\mathbb{F}_1,\mathbb{F}_2) + I(\mathcal{R}_{\mathcal{D}_1}\cup\mathcal{R}_{\mathcal{D}_2};\mathbb{F}_2|\mathbb{F}_1)$, and the fact that $H(\mathcal{R}_{\mathcal{D}_1}\cup\mathcal{R}_{\mathcal{D}_2}|\mathbb{F}_1,\mathbb{F}_2) = 0$, since $\mathcal{R}_{\vec{w}}$ is generated by file-sets $\mathbb{F}_1$ and $\mathbb{F}_2$.

We then bound each of the mutual-information terms in (7). Noting that all files in $\mathbb{F}_1$ can be reconstructed from: (i) the transmissions for the request patterns in $\mathcal{D}_1$, and (ii) the local storages of users in $\mathcal{U}_1$, we have

$$\begin{aligned}
H(\mathbb{F}_1) &= I(\mathbb{F}_1;\mathcal{R}_{\mathcal{D}_1},\mathcal{M}_{\mathcal{U}_1}) \\
&\leq I(\mathbb{F}_1;\mathcal{R}_{\mathcal{D}_1}) + H(\mathcal{M}_{\mathcal{U}_1}),
\end{aligned} \tag{8}$$

where the second equality is due to the chain rule. Since the total cache size $\bigcup_{k\in\mathcal{U}_1}\mathcal{M}_k$ is $s_1\cdot M = \frac{N_1F_1}{2}$, we have

$$I(\mathbb{F}_1;\mathcal{R}_{\mathcal{D}_1}) \geq H(\mathbb{F}_1) - H(\mathcal{M}_{\mathcal{U}_1}) \geq \frac{N_1F_1}{2}. \tag{9}$$

Similarly, we have $I(\mathbb{F}_1;\mathcal{R}_{\mathcal{D}_2}) \geq \frac{N_1F_1}{2}$.

Finally, using a similar logic, the entropy of type-2 files $\mathbb{F}_2$ can be written as

$$\begin{aligned}
H(\mathbb{F}_2) &= I(\mathbb{F}_2;\mathcal{R}_{\mathcal{D}_1}\cup\mathcal{R}_{\mathcal{D}_2},\mathcal{M}_{\mathcal{U}_2}|\mathbb{F}_1) \\
&\leq I(\mathbb{F}_2;\mathcal{R}_{\mathcal{D}_1}\cup\mathcal{R}_{\mathcal{D}_2}|\mathbb{F}_1) + H(\mathcal{M}_{\mathcal{U}_2}).
\end{aligned} \tag{10}$$

Here, the first equality is due to the independence of $\mathbb{F}_1$ and $\mathbb{F}_2$, along with the fact that $\mathbb{F}_2$ can be decoded from $\bigcup_{\vec{w}\in\mathcal{D}_1\cup\mathcal{D}_2}\mathcal{R}_{\vec{w}}$ and $\bigcup_{k\in\mathcal{U}_2}\mathcal{M}_k$.

Since the size of $\bigcup_{k\in\mathcal{U}_2}\mathcal{M}_k$ is $s_2M = \frac{N_2F_2}{2}$, we have

$$I(\mathbb{F}_2;\mathcal{R}_{\mathcal{D}_1}\cup\mathcal{R}_{\mathcal{D}_2}|\mathbb{F}_1) \geq \frac{N_2F_2}{2}. \tag{11}$$

The result of Lemma 2 then follows. ■

We can then easily generalize the above proof for the case of $T \geq 3$ by choosing T groups of users $\mathcal{U}_1$ to $\mathcal{U}_T$, each having $s_l = \frac{N_lF_l}{2M}$ users. This is always possible when **Assumptions 1-2** hold. The rest of the derivation for Proposition 1 follows by applying the above techniques iteratively.

A. Relaxing Assumptions 1 & 2

To relax **Assumptions 1-2**, the analysis is more complicated as we need to consider many corner cases. In the following, we provide the general lower bound without these assumptions and provide a sketch of the proof.

Proposition 2: The worst-case transmission rate is lower bounded by $R^*(\mathbb{F})$, where $R^*(\mathbb{F})$ is the infimum of all values of R that satisfy

$$R > \max\left(\sum_{l=T_1+1}^{T_2-1}\frac{N_lF_l}{2} + \frac{N'_{T_2}F_{T_2}}{2},\ \sum_{l=T_3}^{T_4-1}\frac{N_lF_l^2}{32M'} + \frac{N'_{T_4}F_{T_4}^2}{32M'}\right),$$

where $M' = M - \sum_{l=1}^{T_1}N_l(F_l - R)$, and the parameters T_1 to T_4, N'_{T_2} and N'_{T_4} are of integer values and can be uniquely computed (for any given R) in the following way. (i) T_1 is the largest index l such that $F_l > 2R$. If no such l exists, choose $T_1 = 0$; (ii) T_2 is the largest index satisfying (a) $T \geq T_2 > T_1$, (b) $F_{T_2} > 2M'$, and (c) $\sum_{l=T_1+1}^{T_2-1}N_l < K$. If no such T_2 exists, then choose $T_2 = T_1$ and $N'_{T_2} = 0$. Otherwise, choose $N'_{T_2} = \min(N_{T_2}, K - \sum_{l=T_1+1}^{T_2-1}N_l)$. (iii) T_3 is the smallest index l such that $F_l \leq 2M'$. If no such l exists, then $T_3 = T+1$; (iv) T_4 is the largest index satisfying (a) $T \geq T_4 \geq T_3$, and (b) $\sum_{l=T_3}^{T_4-1}N_lF_l < 2KM'$. If no such T_4 exists, then choose $T_4 = T_3$ and $N'_{T_4} = 0$. Otherwise, choose $N'_{T_4} = \min(N_{T_4}, \lfloor(2KM' - \sum_{l=T_3}^{T_4-1}N_lF_l)/F_{T_4}\rfloor)$.

The main intuition of this general lower bound is as follows. We divide the file types into three groups. Group 1: types 1 to T_1, Group 2: (T_1+1) to T_2, and Group 3: T_3 to T_4. Suppose that there exists a scheme that can achieve a worst-case transmission rate R. It implies that the transmission rate R has to cover all possible request patterns chosen from all three groups of files. Then, we quantify the impact of the requests for each file group and derive the condition on R.

First, note that each file of type l in Group 1 is larger than $2R$. To satisfy all requests for a given file from Group 1, each node needs to store at least $(F_l - R)$ amount of its contents. Therefore, the remaining cache size that can be used to satisfy file requests for Groups 2 and 3 is upper bounded by M'. (This argument can be made precise using conditional entropy.) As a result, when considering Groups 2 and 3, we can treat it equivalently as if the effective cache size has been reduced to M'. We now consider Group 2. We first notice that for any $l = T_1+1$ to T_2, we must have $M' < \frac{F_l}{2}$. We then consider a request pattern in which $\sum_{l=T_1+1}^{T_2-1}N_l + N'_{T_2}$ users request $\sum_{l=T_1+1}^{T_2-1}N_l + N'_{T_2}$ distinct files. By a simple cut-set bound argument, we can obtain $R > \sum_{l=T_1+1}^{T_2-1}\frac{N_lF_l}{2} + \frac{N'_{T_2}F_{T_2}}{2}$.

For Group 3, we use the same bounding techniques as in Prop. 1, i.e., we choose larger files multiple times in the set of request patterns. We can then show that, in order to satisfy the requests for files in Group 3, it requires a minimum rate that is larger than $\sum_{l=T_3}^{T_4-1}\frac{N_lF_l^2}{32M'} + \frac{N'_{T_4}F_{T_4}^2}{32M'}$. Proposition 2 then follows. Note that the rate for files in Group 3 can be compared to (4). It is looser now since we have relaxed **Assumptions 1-2**.

IV. A New Achievable Scheme

In this section, we compare two achievable schemes. One assumes a uniform caching probability. The other adopts a proportional caching probability. It will be shown that the second scheme achieves a lower rate than the first one.

Consider an achievable scheme where the fractions of every file are cached with an equal probability q, as in [1]–[7]. Due to the memory constraint, we have $\sum_{l\in\Phi}qN_lF_l = M$ and $q = \frac{M}{\sum_{l\in\Phi}N_lF_l}$. Using the results in [2], we can show that its achievable rate R_{uni}, with some choice on F_l, N_l and M, can be lower bounded by

$$R_{\text{uni}} \geq \frac{KF_1(1-q)}{1+Kq} \geq \frac{F_1\sum_{l=1}^{T}N_lF_l}{4M}. \tag{12}$$

By choosing $T = \log_2 K$ and $N_{l+1} = 4N_l$, we can show that the lower bound (4) is $\log_2 K \cdot N_1 F_1^2/(4M)$ and the achievable bound (12) is larger than $KN_1F_1^2/(8M)$. The gap between those bounds can be as large as $\Theta(K/\log_2 K)$.

Next, we present the second scheme. Again, we first impose **Assumptions 1-2**. Let $\overline{T} = \min(T, \log_2 K)$. For every file in $\mathbb{F}_l$ ($l \leq \overline{T}$), denote its caching probability as q_l. Namely, each user k will cache $q_l F_l$ of every file in $\mathbb{F}_l$. The overall cache-size constraint thus implies that $\sum_{l=1}^{\overline{T}} q_l N_l F_l = M$. The key difference from the first scheme is that we choose q_l to be linearly proportional to F_l. In other words, the amount of cached content for a file of type l is *quadratically* proportional to F_l. The motivation for this choice of q_l is as follows. Note that if we consider all request patterns where all K users request only the files with size F_l, the rate needed can be approximated[1] by $\frac{F_l}{q_l}$. Thus, in order to minimize the worst-case value, i.e., $\max_l \frac{F_l}{q_l}$, we should choose q_l to be proportional to the file size F_l, i.e., $q_l \triangleq QF_l$ where the constant Q equals to $M/(\sum_{l=1}^{\overline{T}} N_l F_l^2)$. This choice is also consistent with the choice of request patterns $\mathcal{D}$ in the proof of Lemma 2 and Proposition 1. Since the larger files in $\mathbb{F}_1$ is requested twice as frequently as the shorter files in $\mathbb{F}_2$, it suggests that more cache space may be allocated to $\mathbb{F}_1$.

We now describe the transmission design. **Initialization:** each user k reconstructs the portion of the requested file $\mathcal{F}_{w_k}$ that is stored in its local cache $\mathcal{M}_k$. **Transmission:** for any non-empty subset $\mathcal{U} \subset \{1, \cdots, K\}$, we do the following. For each $k \in \mathcal{U}$, we assemble the portion of the requested file $\mathbb{F}_{w_k}$ that is cached by all users $h \in \mathcal{U}\backslash k$ but not by user k as a continuous bit-string and denote the assembled bit-string by B_k. Then, we send the bit-wise XORed[2] string $\bigoplus_{k\in\mathcal{U}} B_k$. Note that some bit string B_k may be shorter than the other $B_{k'}$. We simply zero-padded the shorter bit strings during XOR. In the end, the total amount of bits sent for a given $\mathcal{U}$ is $\max_{k\in\mathcal{U}} |B_k|$. After finishing transmission for all $\mathcal{U}$, it is guaranteed that all users can recover the desired packets.

Note that in our construction, those files of type $l > \overline{T}$ will never be cached. As a result, if any user k requests such a file, the entire file will be treated as a single bit-string and transmitted separately. By analyzing the bit-length of each transmission $\mathcal{U}$, we can upper bound the transmission rate by

$$R_{\text{prop}} \triangleq \sum_{\mathcal{U}} \max_{k\in\mathcal{U}} |B_k| \leq (\overline{T}+1) \sum_{l=1}^{\overline{T}} N_l F_l^2/M. \tag{13}$$

Comparing Eqs. (13) and (4), the gap is at most $4(\log_2 K+1)$, which is a dramatic improvement from (12). Thus, using a caching probability q_l proportional to F_l is critical.

Note that while our scheme allows coding across different file-types, the inequality in (13) does not exploit this gain, which may be the reason for the $\Theta(\log_2 K)$-factor gap between our upper and lower bounds. For future work, it would be interesting to see whether this gap can be removed.

[1]This approximation can be observed from the first inequality in (12).

[2]If ≥ 2 users request the same file, then we only XOR the string once. The reason is that XOR the same string twice will give a zero-string.

When relaxing **Assumptions 1-2**, the analysis becomes more complicated due to many corner cases. We can have the following result.

Proposition 3: We can construct a modified scheme that achieves $R_{\text{prop}} \leq (32\log_2 K + 22)R^*(\mathbb{F})$, where $R^*(\mathbb{F})$ is specified in Proposition 2.

Namely, the gap to the lower bound is at most $\Theta(\log_2 K)$.

V. Numerical Comparison

We compare the two lower bounds, (3) and (4), and the two upper bounds, (12) and (13), in the following two numeric examples. There are $K = 8$ users, each with a cache size $M = 128$. System 1 has 2 file types with $F_1 = 8$, $N_1 = 16$, $F_2 = 4$, and $N_2 = 64$. System 2 has 3 file types with $F_1 = 8$, $N_1 = 16$, $F_2 = 4$, $N_2 = 64$, $F_3 = 2$, and $N_3 = 128$. For the achievable schemes, instead of deriving the bounds, we list the exact R_{prop} (UB2 in Table I) and R_{uni} (UB1 in Table I) values by numerically computing $\sum_{\mathcal{U}} \max_{k\in\mathcal{U}} |B_k|$. These numerical results verify our findings, i.e., not only the proposed lower bound (4) is greater than the result in (3) that uses the traditional cut-set bounds, but also R_{prop} is much larger than R_{uni}. That is, proportional caching probability significantly outperforms uniform caching probability.

VI. Conclusion

In this paper, we study coded caching for systems where files of interest are of different sizes. We provide tighter lower-bound and achievable bound for the worst-case transmission rate, which differ by at most a $\Theta(\log K)$ factor. The key novelty is a new cut-set (lower) bound that considers request patterns where larger files are requested more times.

Acknowledgment

This work was partially supported by NSF grants: CCF-0845968, ECCS-1407603, and CCF-1422997, a grant from the Army Research Office W911NF-14-1-0368, and two grants from NSF China (No. 61325012, 61271219).

References

[1] M.A. Maddah-Ali, and U. Niesen, "Fundamental Limits of Caching", in *IEEE Trans. Inform. Theory*, vol. 60, no. 5, pp. 2856-2867, May 2014.

[2] M.A. Maddah-Ali, and U. Niesen, "Decentralized Coded Caching Attains Order-Optimal Memory-Rate Tradeoff", to appear in *IEEE/ACM Trans. Netw.*, 2014.

[3] N. Karamchandani, U. Niesen, M.A. Maddah-Ali and S. Diggavi, "Hierarchical Coded Caching", *arXiv:1403.7007v2 [cs.IT]*, Jun. 2014.

[4] J. Hachem, N. Karamchandani and S. Diggavi, "Multi-level Coded Caching", *arXiv:1404.6563 [cs.IT]*, Apr. 2014.

[5] U. Niesen, and M.A. Maddah-Ali, "Coded Caching with Nonuniform Demands", *arXiv:1308.0178v2 [cs.IT]*, Mar. 2014.

[6] M. Ji, A. Tulino, J. Llorca and G. Caire, "On the Average Performance of Caching and Coded Multicasting with Random Demands", *arXiv:1402.4576v2 [cs.IT]*, Jul. 2014.

[7] R. Pedarsani, M.A. Maddah-Ali and U. Niesen, "Online Coded Caching", *arXiv:1311.3646 [cs.IT]*, Nov. 2013.

[8] R.W. Yeung, "A Framework for Linear Information Inequalities", in *IEEE Trans. Inform. Theory*, vol. 43, no. 6, pp. 1924-1934, Nov. 1997.