Closing the Gap for Coded Caching with Distinct File Sizes

Jinbei Zhang[†‡], Xiaojun Lin[§], Chih-Chun Wang[§]
[†]School of Electronic and Communication Engineering, Sun Yat-sen University, China
[‡]The State Key Laboratory of Integrated Services Networks, Xidian University, China
[§]School of Electrical and Computer Engineering, Purdue University, USA
Email: zhjinbei@mail.sysu.edu.cn, {linx,chihw}@purdue.edu

***Abstract*—Coded caching can exploit multicast opportunities even when multiple users request different pieces of content, and thus can significantly reduce the backhaul requirement to serve high-volume content. A common assumption in existing studies of coded caching is that all files are with the same size, which however may not be true in reality. Our previous work [1] first studied this problem, and proposed a non-trivial lower bound, as well as a new achievable scheme that uses a caching probability increasing proportionally with the file size. However, the gap [1] of the achievable rate and the lower bound still differs by a factor of $\Theta(\log K)$, where K is the number of users in the system. In this paper, under a mild assumption that the total size of all files is larger than eight times the size of one individual cache, we will close this gap and reduce it to a constant by proposing a novel new lower bound and another new achievable scheme. Our lower bound is derived by considering a new cut-set bound, where files of a type[1] will be requested more often if the number of such files is smaller. Our achievable scheme uses a caching probability that decreases with the number of files with a same type. The improvements on both the lower bound and the achievable rate make their gap constant.**

I. Introduction

Coded caching [2] has received intense attention recently. The general setting is to let multiple users requesting files from a same server through a shared broadcast link. By exploiting the usage of user caches, coded caching can significantly reduce the overall rate needed in the broadcast link compared to traditional uncoded schemes. The transmission procedure is composed by two phases, i.e., placement phase and delivery phase. In the placement phase, the server has no knowledge on the users' requests, and will send files (or subfiles) to fill up the user caches during non-peak hours. In the delivery phase, users' requests are revealed, and the server will send whatever contents needed to satisfy users' requests, with knowledge on users' cached content placed in the placement phase.

Traditional uncoded schemes may fill users' caches with the same contents in the placement phase. Thus, during the delivery phase, if users' request are distinct, the overall rate needed can be seen as the summation of rate needed by each user. Compared to no caches at users, the gain brought by users' caches is only related to the size of the local cache of each user, which is called local caching gain in [2]. Different with traditional schemes, coded caching may fill up the users' caches with different contents. Then, during the delivery phase, for a user group, the content requested by each user may be cached in every other users of the same group. An XOR signal between these requested contents can be sent to benefit all users in the group. This multicast gain is called global caching gain [2]. Using a simple cut-set argument, the worst-case performance of this novel coded caching scheme is shown to be a constant gap from the information theoretical converse.

Along this line, there are many recent works. [3] extends the centralized coded scheme into a decentralized one, where each user places its cache with a random portion of each file. [4] studies hierarchical networks where end users may not directly connect to the server, but instead through relay nodes. When files have different popularity, the average rate is investigated in [5]–[7]. In all of these studies, a common theme is to assume that all files are with the same size. Thus, the caching and transmission scheme is independent of the file sizes.

However, in reality, it is common that files are of significantly varied sizes. Several questions arise naturely. The first question is to ask, if some files are larger, whether a larger portion of them should be cached during the placement phase. Further, the contents to be XORed in the delivery phase may not be of the same size either. Thus, another question is to ask whether these transmissions indicate some "wastes" on multicast opportunities. For the first question, our previous work [1] proposed an achievable scheme that employs a caching probability increasing proportional to file sizes. We showed that it performs much better than the scheme employing a uniform caching probability. To evaluate the second question, a information theoretical converse is needed, which is more involved. In [1], the converse is derived by considering a new cut-set bound where larger files are requested more times. However, the gap between the achievable rate and the corresponding converse is up to $\Theta(\log K)$. It is unclear whether the converse is loose or the transmission scheme wastes some multicast opportunities.

In this paper, we continue to investigate this problem and will reduce the gap from $\Theta(\log K)$ to a constant, when the total size of all files is larger than eight times the size of one individual cache. The main contributions of this paper are two fold. First, we propose a new lower bound, where files of a type will be requested more often if the number of such files is smaller. Second, we propose a new achievable scheme, which uses a caching probability that decreases with the number of

[1]A file type can be seen as files with comparable file sizes.

files with same type. We will show that, in some settings, the new converse obtained in this paper can be $\Theta(\log K)$ times larger than the lower bound in [1], while the proposed achievable rate remains the same as that in [1]. In some other settings, the proposed achievable rate can reduce the rate in [1] by a $\Theta(\log K)$ factor, while the proposed lower bound remains the same as that in [1]. As a result, both improvements on the lower bound and the achievable rate make the gap constant.

The rest of the paper is organized as follows. We present the network model in Section II. In Section III, the lower bounds of the worst-case transmission rate are derived. The achievable rate and comparisons are conducted in Section IV. Finally, we conclude.

II. Network Model

In general, files can have arbitrary sizes. Without loss of generality, one can normalize the file sizes into ones that differ by power-of-2 factors. Under mild conditions [1], the rates needed before and after the file size normalization will not differ by more than a constant. Thus, in this paper we will only consider file systems with the above property.

Assume that the file set is $\mathbb{F}$, and there are at most T distinct file sizes. We now call the files of the same size as "of the same type". Thus, we introduce the following 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 $\mathbb{F}_l$, denoted by $\mathbb{F}_l = \{\mathcal{F}_{lj} | 1 \leq j \leq N_l\}$.

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 $\mathcal{C}_k$. We assume that $\sum_{l=1}^{T} N_l F_l \geq 8M$ and $\sum_{l=1}^{T} N_l \geq K$, 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 $\mathcal{C}_k$, with the hope of reducing the rate needed to satisfy users' requests during peak hours. This process is called the placement phase. We emphasize that the placement phase must be completed before any file requests are made.

During peak hours, the k-th user will request a file in $\mathbb{F}$, 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 $\mathcal{C}_k$, 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 $\mathcal{C}_k$. 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 $\mathcal{C}_k, k = 1, ..., K$; 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 $\varepsilon > 0$, there exists a caching and transmission scheme such that, for large enough file sizes, 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 $R(\mathbb{F})$.

III. Lower Bounds of the Worst-case Transmission Rate

In this section, we present two lower bounds for the worst-case transmission rate. We first present the existing lower bound, i.e., Proposition 1 in [1], as follows.

Lemma 1: When $\frac{2M}{F_1}$ and $\frac{N_l F_l}{2M}$ are both integers for all file types l and $\sum_{l=1}^{\min(T,\log_2 K)} \frac{N_l F_l}{2M} \leq K$, the worst-case transmission rate can be lower bounded by

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

In [1], it is also shown that the achievable rate is at most a $\Theta(\log_2 K)$ factor higher than the lower bound presented in Lemma 1.

We next present the second lower bound, which will be shown to be order optimal. For ease of presentation, we first focus on the case when the following two assumptions hold.

Assumption 1: $N_2 = 2N_1$, and $N_l = 4^{l-2}N_2$, for all types l such that $2 \leq l \leq \overline{T}$, where $\overline{T}$ is the maximum integer[2] that satisfies $2^{\overline{T}-1} \cdot \frac{(\overline{T}+1)N_1F_1}{4M} \leq K$.

Assumption 2: $\frac{4M}{(\overline{T}+1)F_1}$ and $\frac{(\overline{T}+1)N_1F_1}{4M}$ are both integers.

These two assumptions simplify our exposition below because we will not need to be concerned with corner cases, or use floor and ceiling functions. While these two assumptions simplify the analysis of Proposition 1, the key insight of our contributions and analysis still hold under general settings, as we will show in Section IV when we relax these two assumptions.

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

$$R^*(\mathbb{F}) \geq \frac{\left(\sum_{i=1}^{\overline{T}} 2^{1-i} H(\mathbb{F}_i)\right)^2}{4N_1M}. \tag{2}$$

To compare (1) and (2), let $\overline{T} = \log_2 K$. Note that $F_{i+1} = \frac{1}{2}F_i$ and $H(\mathbb{F}_i) = N_iF_i$. Along with **Assumption 1**, the RHS of (1) equals to $(\log K + 1)N_1F_1^2/(8M)$. On the other hand, the RHS of (2) equals to $(\log_2 K + 1)^2 N_1F_1^2/(16M)$, which is a $\Theta(\log_2 K)$ factor higher than (1). Thus, if the same scheme in [1] is employed, the gap between the achievable rate and the RHS of (2) must be bounded as a constant.

To begin with, we next focus on the simpler case when there are only two file types, i.e., $T = 2$. While this case is simpler, the key ideas remain the same and the differences with [1] will be revealed.

[2]When all users request files from $\mathbb{F}_l$ ($l > \overline{T}$), the lower bound will not change since these files can be seen as too small to impact the lower bound.

To derive the new lower bound, we construct a new set of request patterns that differs from [1], as illustrated in Fig. 1. We choose two independent user sets $\mathcal{U}_1$ and $\mathcal{U}_2$ of size s each, i.e., $|\mathcal{U}_1| = |\mathcal{U}_2| = s$, where s is defined later, and is an integer with **Assumptions** 1-2. We divide file set $\mathbb{F}_2$ into two disjoint subsets with the same size, i.e., $\mathbb{F}_{21}$ and $\mathbb{F}_{22}$.

We now explain the construction of request patterns $\mathcal{D}_1$ and $\mathcal{D}_2$. In the very first request pattern $\vec{w}$ in $\mathcal{D}_1$, let the s users in $\mathcal{U}_1$ request the first s files in $\mathbb{F}_1$, and let the s users in $\mathcal{U}_2$ request the first s files in $\mathbb{F}_{21}$. Then, in the second request $\vec{w}$ in $\mathcal{D}_1$, let the s users in $\mathcal{U}_1$ requests the second s files in $\mathbb{F}_1$ and let the s users in $\mathcal{U}_2$ request the second s files in $\mathbb{F}_{21}$. Continue this construction until $|\mathcal{D}_1| = N_1/s$, i.e., each of the N_1 files in $\mathbb{F}_1$ has all been requested once. At the same time, since $N_2 = 2N_1$ and $|\mathcal{U}_1| = |\mathcal{U}_2|$, the first half files in $\mathbb{F}_2$, denoted as $\mathbb{F}_{21}$, has 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 $\mathbb{F}_{22}$.

We then swap $\mathbb{F}_1$ and $\mathbb{F}_2$ in the rest of the request patterns. Specifically, in $\mathcal{D}_3$, the users in $\mathcal{U}_1$ will request the files in $\mathbb{F}_{21}$, while the users in $\mathcal{U}_2$ will request the files in $\mathbb{F}_1$. In $\mathcal{D}_4$, the users in $\mathcal{U}_1$ will request the files in $\mathbb{F}_{22}$ instead, while the users in $\mathcal{U}_2$ still request the files in $\mathbb{F}_1$.

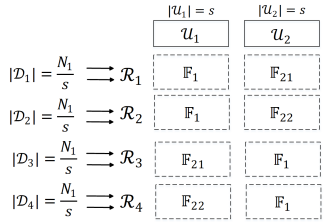

Fig. 1: An illustration on the request patterns for a file system with two types.

Remark: There are two main differences between the request patterns in this paper and that in [1]. First, there are only two request batches, i.e., $\mathcal{D}_1$ and $\mathcal{D}_2$ in [1], while there are four batches in this paper. Second, the number of selected users in $\mathcal{U}_1$ is given as $|\mathcal{U}_1| = \frac{N_1 F_1}{2M}$ and is not required to equal to $|\mathcal{U}_2|$. Thus, we can not swap $\mathcal{U}_1$ and $\mathcal{U}_2$ to construct $\mathcal{D}_3$ and $\mathcal{D}_4$ in Fig. 1. In this paper, the number of selected users in each user set is the same, i.e., s, but is a variable that can be optimized.

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, as specified in Lemma 2. For ease of presentation, we denote $\mathcal{R}_i \triangleq \bigcup_{\vec{w}\in\mathcal{D}_i} \mathcal{R}_{\vec{w}}$ for $i = 1, 2, 3, 4$, and $\mathcal{M}_j \triangleq \bigcup_{k\in\mathcal{U}_j} \mathcal{C}_k$ for $j = 1, 2$.

Lemma 2: For the request patterns constructed in Figure 1, the worst-case rate should satisfy

$$\begin{aligned}\sum_{i=1}^{4} H(\mathcal{R}_i) \geq & 4H(\mathbb{F}_1) + 2H(\mathbb{F}_2) - 2H(\mathcal{M}_1) - 2H(\mathcal{M}_2) \\ & + H(\mathcal{R}_1, \mathcal{R}_2|\mathbb{F}_1, \mathbb{F}_2) + H(\mathcal{R}_3, \mathcal{R}_4|\mathbb{F}_1, \mathbb{F}_2) \\ & + H(\mathcal{M}_1, \mathcal{M}_2|\mathbb{F}_1, \mathbb{F}_2, \mathcal{R}_1, \mathcal{R}_2) \\ & + H(\mathcal{M}_1, \mathcal{M}_2|\mathbb{F}_1, \mathbb{F}_2, \mathcal{R}_3, \mathcal{R}_4).\end{aligned} \tag{3}$$

Proof: Summing over all the rates and conditioning them on $\mathbb{F}_1$, we have

$$\begin{aligned}\sum_{i=1}^{4} H(\mathcal{R}_i) &= \sum_{i=1}^{4} H(\mathcal{R}_i|\mathbb{F}_1) + \sum_{i=1}^{4} I(\mathcal{R}_i; \mathbb{F}_1) \\ &= H(\mathcal{R}_1, \mathcal{R}_2|\mathbb{F}_1) + I(\mathcal{R}_1; \mathcal{R}_2|\mathbb{F}_1) \\ &\quad + H(\mathcal{R}_3, \mathcal{R}_4|\mathbb{F}_1) + I(\mathcal{R}_3; \mathcal{R}_4|\mathbb{F}_1) + \sum_{i=1}^{4} I(\mathcal{R}_i; \mathbb{F}_1).\end{aligned} \tag{4}$$

We then quantify each term in (4). First, for $H(\mathcal{R}_1, \mathcal{R}_2|\mathbb{F}_1)$, we have

$$\begin{aligned}&H(\mathcal{R}_1, \mathcal{R}_2|\mathbb{F}_1) + H(\mathcal{M}_2|\mathbb{F}_1) \\ &= H(\mathcal{R}_1, \mathcal{R}_2|\mathbb{F}_1, \mathbb{F}_2) + I(\mathcal{R}_1, \mathcal{R}_2; \mathbb{F}_2|\mathbb{F}_1) + H(\mathcal{M}_2|\mathbb{F}_1) \\ &= H(\mathcal{R}_1, \mathcal{R}_2|\mathbb{F}_1, \mathbb{F}_2) + I(\mathcal{R}_1, \mathcal{R}_2, \mathcal{M}_2; \mathbb{F}_2|\mathbb{F}_1) \\ &\quad - I(\mathcal{M}_2; \mathbb{F}_2|\mathcal{R}_1, \mathcal{R}_2, \mathbb{F}_1) + H(\mathcal{M}_2|\mathbb{F}_1) \\ &\geq H(\mathcal{R}_1, \mathcal{R}_2|\mathbb{F}_1, \mathbb{F}_2) + I(\mathcal{R}_1, \mathcal{R}_2, \mathcal{M}_2; \mathbb{F}_2|\mathbb{F}_1) \\ &\quad + H(\mathcal{M}_2|\mathcal{R}_1, \mathcal{R}_2, \mathbb{F}_1, \mathbb{F}_2) \\ &= H(\mathcal{R}_1, \mathcal{R}_2|\mathbb{F}_1, \mathbb{F}_2) + H(\mathbb{F}_2) + H(\mathcal{M}_2|\mathcal{R}_1, \mathcal{R}_2, \mathbb{F}_1, \mathbb{F}_2).\end{aligned} \tag{5}$$

Here, the first and third equalities follow from the definition and basic properties of entropy. The second equality is due to the chain rule for mutual information. For the fourth equality, noting that all files in $\mathbb{F}_2$ can be reconstructed from: (i) the transmissions for the request patterns in $\mathcal{D}_1$ and $\mathcal{D}_2$, and (ii) the local storages of users in $\mathcal{U}_2$. Hence, we have $I(\mathcal{R}_1, \mathcal{R}_2, \mathcal{M}_2; \mathbb{F}_2|\mathbb{F}_1) \geq I(\mathbb{F}_2; \mathbb{F}_2|\mathbb{F}_1) = H(\mathbb{F}_2)$, where the equality is due to the independence of $\mathbb{F}_1$ and $\mathbb{F}_2$. On the other hand, $I(\mathcal{R}_1, \mathcal{R}_2, \mathcal{M}_2; \mathbb{F}_2|\mathbb{F}_1) \leq H(\mathbb{F}_2)$. Therefore, we have the fourth equality in (5).

For $H(\mathcal{R}_3, \mathcal{R}_4|\mathbb{F}_1)$, similar to (5), we have

$$\begin{aligned}&H(\mathcal{R}_3, \mathcal{R}_4|\mathbb{F}_1) + H(\mathcal{M}_1|\mathbb{F}_1) \\ &\geq H(\mathcal{R}_3, \mathcal{R}_4|\mathbb{F}_1, \mathbb{F}_2) + H(\mathbb{F}_2) + H(\mathcal{M}_1|\mathcal{R}_3, \mathcal{R}_4, \mathbb{F}_1, \mathbb{F}_2).\end{aligned} \tag{6}$$

Now we quantify $I(\mathcal{R}_i; \mathbb{F}_1)$ for $i = 1, 2, 3, 4$ in (4). Since $\mathbb{F}_1$ can be decoded from $\mathcal{R}_1$ and $\mathcal{M}_1$,

$$\begin{aligned}H(\mathbb{F}_1) &= I(\mathcal{R}_1, \mathcal{M}_1; \mathbb{F}_1) \\ &= I(\mathcal{R}_1; \mathbb{F}_1) + I(\mathcal{M}_1; \mathbb{F}_1|\mathcal{R}_1),\end{aligned} \tag{7}$$

where the second equality is due to the chain rule. Conditioning $\mathcal{M}_1$ on R_1, we have

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

With equalities (7) and (8), we can obtain

$$I(\mathcal{R}_1; \mathbb{F}_1) = H(\mathbb{F}_1) - H(\mathcal{M}_1) + H(\mathcal{M}_1|\mathbb{F}_1, \mathcal{R}_1) + I(\mathcal{M}_1; \mathcal{R}_1). \tag{9}$$

Similarly, we have

$$I(\mathcal{R}_2; \mathbb{F}_1) = H(\mathbb{F}_1) - H(\mathcal{M}_1) + H(\mathcal{M}_1|\mathbb{F}_1, \mathcal{R}_2) + I(\mathcal{M}_1; \mathcal{R}_2). \tag{10}$$

$$I(\mathcal{R}_3; \mathbb{F}_1) = H(\mathbb{F}_1) - H(\mathcal{M}_2) + H(\mathcal{M}_2|\mathbb{F}_1, \mathcal{R}_3) + I(\mathcal{M}_2; \mathcal{R}_3). \tag{11}$$

$$I(\mathcal{R}_4; \mathbb{F}_1) = H(\mathbb{F}_1) - H(\mathcal{M}_2) + H(\mathcal{M}_2|\mathbb{F}_1, \mathcal{R}_4) + I(\mathcal{M}_2; \mathcal{R}_4). \tag{12}$$

Summing over the third term in the RHS of (11), the third term in the RHS of (12), and $I(\mathcal{R}_3;\mathcal{R}_4|\mathbb{F}_1)$ in the RHS of (4), we can prove that[3]

$$\begin{aligned} &H(\mathcal{M}_2|\mathbb{F}_1,\mathcal{R}_3)+H(\mathcal{M}_2|\mathbb{F}_1,\mathcal{R}_4)+I(\mathcal{R}_3;\mathcal{R}_4|\mathbb{F}_1) \\ &\quad\geq H(\mathcal{M}_2|\mathbb{F}_1)+H(\mathcal{M}_2|\mathbb{F}_1,\mathbb{F}_2,\mathcal{R}_3,\mathcal{R}_4). \end{aligned} \tag{13}$$

Similar to (13), summing over the third term in the RHS of (9), the third term in the RHS of (10), and $I(\mathcal{R}_1;\mathcal{R}_2|\mathbb{F}_1)$ in the RHS of (4), we have

$$\begin{aligned} &H(\mathcal{M}_1|\mathbb{F}_1,\mathcal{R}_1)+H(\mathcal{M}_1|\mathbb{F}_1,\mathcal{R}_2)+I(\mathcal{R}_1;\mathcal{R}_2|\mathbb{F}_1) \\ &\quad\geq H(\mathcal{M}_1|\mathbb{F}_1)+H(\mathcal{M}_1|\mathbb{F}_1,\mathbb{F}_2,\mathcal{R}_1,\mathcal{R}_2). \end{aligned} \tag{14}$$

Now we are ready to prove the result of this lemma.

$$\begin{aligned} \sum_{i=1}^{4}H(\mathcal{R}_i) &= (5)-H(\mathcal{M}_2|\mathbb{F}_1)+I(\mathcal{R}_1;\mathcal{R}_2|\mathbb{F}_1) \\ &\quad+(6)-H(\mathcal{M}_1|\mathbb{F}_1)+I(\mathcal{R}_3;\mathcal{R}_4|\mathbb{F}_1) \\ &\quad+(9)+(10)+(11)+(12) \\ &\geq 4H(\mathbb{F}_1)+2H(\mathbb{F}_2)-2H(\mathcal{M}_1)-2H(\mathcal{M}_2) \\ &\quad+H(\mathcal{R}_1,\mathcal{R}_2|\mathbb{F}_1,\mathbb{F}_2)+H(\mathcal{M}_2|\mathcal{R}_1,\mathcal{R}_2,\mathbb{F}_1,\mathbb{F}_2)-H(\mathcal{M}_1|\mathbb{F}_1) \\ &\quad+H(\mathcal{R}_3,\mathcal{R}_4|\mathbb{F}_1,\mathbb{F}_2)+H(\mathcal{M}_1|\mathcal{R}_3,\mathcal{R}_4,\mathbb{F}_1,\mathbb{F}_2)-H(\mathcal{M}_2|\mathbb{F}_1) \\ &\quad+H(\mathcal{M}_1|\mathbb{F}_1,\mathcal{R}_1)+H(\mathcal{M}_1|\mathbb{F}_1,\mathcal{R}_2)+I(\mathcal{R}_1;\mathcal{R}_2|\mathbb{F}_1) \\ &\quad+H(\mathcal{M}_2|\mathbb{F}_1,\mathcal{R}_3)+H(\mathcal{M}_2|\mathbb{F}_1,\mathcal{R}_4)+I(\mathcal{R}_3;\mathcal{R}_4|\mathbb{F}_1) \\ &\geq 4H(\mathbb{F}_1)+2H(\mathbb{F}_2)-2H(\mathcal{M}_1)-2H(\mathcal{M}_2) \\ &\quad+H(\mathcal{R}_1,\mathcal{R}_2|\mathbb{F}_1,\mathbb{F}_2)+H(\mathcal{R}_3,\mathcal{R}_4|\mathbb{F}_1,\mathbb{F}_2) \\ &\quad+H(\mathcal{M}_1|\mathbb{F}_1,\mathbb{F}_2,\mathcal{R}_1,\mathcal{R}_2)+H(\mathcal{M}_2|\mathbb{F}_1,\mathbb{F}_2,\mathcal{R}_1,\mathcal{R}_2) \\ &\quad+H(\mathcal{M}_1|\mathbb{F}_1,\mathbb{F}_2,\mathcal{R}_3,\mathcal{R}_4)+H(\mathcal{M}_2|\mathbb{F}_1,\mathbb{F}_2,\mathcal{R}_3,\mathcal{R}_4) \\ &\geq 4H(\mathbb{F}_1)+2H(\mathbb{F}_2)-2H(\mathcal{M}_1)-2H(\mathcal{M}_2) \\ &\quad+H(\mathcal{R}_1,\mathcal{R}_2|\mathbb{F}_1,\mathbb{F}_2)+H(\mathcal{M}_1,\mathcal{M}_2|\mathbb{F}_1,\mathbb{F}_2,\mathcal{R}_1,\mathcal{R}_2) \\ &\quad+H(\mathcal{R}_3,\mathcal{R}_4|\mathbb{F}_1,\mathbb{F}_2)+H(\mathcal{M}_1,\mathcal{M}_2|\mathbb{F}_1,\mathbb{F}_2,\mathcal{R}_3,\mathcal{R}_4). \end{aligned}$$

Here, the first equality is from (4). The second inequality is obtained by substituting the corresponding equations. The third inequality is due to (13) and (14). ■

Now we illustrate the difference between the analysis in [1] and that in this paper. When $N_2 = 2N_1$, if we directly apply the lower bound in [1] where the traffic patterns only include $\mathcal{D}_1$ and $\mathcal{D}_2$, we would obtain

$$H(\mathcal{R}_1)+H(\mathcal{R}_2)\geq 2H(\mathbb{F}_1)+H(\mathbb{F}_2)-2H(\mathcal{M}_1)-H(\mathcal{M}_2). \tag{15}$$

If we multiple both sides of (15) and compare it with (3), the lower bound in Lemma 2 is increased by another term $2H(\mathcal{M}_1)$. This increase is the key step towards the tighter lower bound in Proposition 1. Indeed, substituting $s=\frac{N_1F_1}{2M}$ and noting that $F_1 = 2F_2$, the $T = 2$ case of Proposition 1 follows immediately from Lemma 2.

For the cases when there are $T \geq 3$ file types and **Assumptions** 1-2 hold, we choose $2^{\overline{T}-1}$ user sets, each having the same number of users, i.e., s. An example to construct request patterns for $\overline{T} = 3$ is given in Table I, where $\mathbb{F}_3$ is divided into eight subsets $\mathbb{F}_{3i}$ $(i = 1, 2, ..., 8)$. Then by applying the intuition and techniques in Lemma 2 iteratively,

[3]Due to space limitation, the proof of (13) is omitted here. One can easily verify this inequality with Xitip [8].

TABLE I: An illustration on the request patterns for a file system with three types

	$\mathcal{M}_1$	$\mathcal{M}_2$	$\mathcal{M}_3$	$\mathcal{M}_4$
$R_1=\frac{N_1}{s}R$	$\mathbb{F}_1$	$\mathbb{F}_{21}$	$\mathbb{F}_{31}$	$\mathbb{F}_{32}$
$R_2=\frac{N_1}{s}R$	$\mathbb{F}_1$	$\mathbb{F}_{22}$	$\mathbb{F}_{33}$	$\mathbb{F}_{34}$
$R_3=\frac{N_1}{s}R$	$\mathbb{F}_{21}$	$\mathbb{F}_1$	$\mathbb{F}_{35}$	$\mathbb{F}_{36}$
$R_4=\frac{N_1}{s}R$	$\mathbb{F}_{22}$	$\mathbb{F}_1$	$\mathbb{F}_{37}$	$\mathbb{F}_{38}$
$R_5=\frac{N_1}{s}R$	$\mathbb{F}_{31}$	$\mathbb{F}_{32}$	$\mathbb{F}_1$	$\mathbb{F}_{21}$
$R_6=\frac{N_1}{s}R$	$\mathbb{F}_{33}$	$\mathbb{F}_{34}$	$\mathbb{F}_1$	$\mathbb{F}_{22}$
$R_7=\frac{N_1}{s}R$	$\mathbb{F}_{35}$	$\mathbb{F}_{36}$	$\mathbb{F}_{21}$	$\mathbb{F}_1$
$R_8=\frac{N_1}{s}R$	$\mathbb{F}_{37}$	$\mathbb{F}_{38}$	$\mathbb{F}_{22}$	$\mathbb{F}_1$

we can obtain

$$\sum_{i=1}^{2^{\overline{T}}}H(\mathcal{R}_i)\geq\sum_{i=1}^{\overline{T}}2^{\overline{T}+1-i}H(\mathbb{F}_i)-\sum_{i=1}^{2^{\overline{T}-1}}2H(\mathcal{M}_i). \tag{16}$$

By letting $s=\frac{\sum_{i=1}^{\overline{T}}2^{1-i}H(\mathbb{F}_i)}{2M}$, we have Proposition 1. Since we need to choose $2^{\overline{T}-1}s$ users, we have the constraint on $\overline{T}$ and integer constraints in **Assumptions** 1-2.

Observation 1: Note that even when the files in $\mathbb{F}_i$ are not of the same size, as long as $|\mathbb{F}_i| = 2\cdot 4^{i-2}N_1$ for $\overline{T}\geq i\geq 2$, we can still construct request patterns as above and apply the same proof. Therefore, Lemma 2 and Proposition 1 still hold, which will be useful for relaxing **Assumptions** 1-2 later.

IV. A New Achievable Scheme and Performance Comparison

As introduced in Section III, under **Assumptions** 1-2, our proposed lower bound can be $\Theta(\log K)$ times that in [1], and thus we can use the scheme in [1] to show a constant gap. However, when **Assumptions** 1-2 do not hold, the improvement of the lower bound may be limited, and a new scheme is required to achieve the constant gap.

Our proposed scheme works as follows. During the placement phase, each user k will cache q_lF_l bits of every file in the l-th file group, where q_l will be specified later. During the delivery phase, the coded caching scheme in [3] is employed to satisfy the users requesting files from the same group.

Different from the scheme in [1], which caches each bit in $\mathbb{F}_l$ with a higher probability if F_l is large, our new scheme will cache each bit in $\mathbb{F}_l$ with a higher probability when N_l is small. We choose $q_l = 0$ when $l > \overline{T}$, where $\overline{T}$ is the maximum integer that satisfies $2^{\overline{T}-1}\cdot\sum_{i=1}^{\overline{T}}2^{1-i}H(\mathbb{F}_i)/(2M)\leq K$. Note that the same $\overline{T}$ was used in the construction of request patterns in Section III. For $l\leq\overline{T}$, we choose q_l to be proportional to $1/\sqrt{N_l}$, i.e., $q_l\triangleq Q/\sqrt{N_l}$ where the constant $Q=M/(\sum_{i=1}^{\overline{T}}\sqrt{N_iF_i})$ is the normalization factor.

The main idea of choosing such q_l is to minimize the approximate rate of our proposed scheme. For a request pattern, if there are K_l users requesting distinct files in $\mathbb{F}_l$, the transmission rate to satisfy the K_l users can be upper bounded as $\frac{K_lF_l(1-q_l)}{(1+K_lq_l)}$ [3]. When $K_l\geq 1/q_l$, the rate $\frac{K_lF_l(1-q_l)}{(1+K_lq_l)}$ can then be approximated (and upper bounded) as $\frac{F_l}{q_l}$. The sum

rate to satisfy users requesting files $\mathbb{F}_l$ ($l \leq \overline{T}$) can then be upper bounded as $\sum_{l=1}^{\overline{T}} \frac{F_l}{q_l}$. In order to minimize this worst-case value under the memory constraint $\sum_{l=1}^{\overline{T}} q_l N_l F_l = M$, we thus should choose q_l to be proportional to $1/\sqrt{N_l}$.

Note that in our construction, those files of type $l > \overline{T}$ will never be cached. As a result, if any user requests such a file, the entire file will be transmitted directly. The overall transmission rate can then be upper bounded as

$$R_{\text{prop}} \leq \sum_{l=1}^{\overline{T}} \frac{F_l}{q_l} + KF_{\overline{T}+1} = \frac{\left(\sum_{l=1}^{\overline{T}} \sqrt{N_l} F_l\right)^2}{M} + KF_{\overline{T}+1}. \quad (17)$$

Under **Assumptions** 1-2, we have $R_{\text{prop}} \leq \frac{\overline{T}^2 N_1 {F_1}^2}{M} + \frac{KF_1}{2^{\overline{T}}} \leq 2\frac{\overline{T}^2 N_1 {F_1}^2}{M}$. On the other hand, the lower bound in (2) equals to $\frac{(\overline{T}+1)^2 N_1 {F_1}^2}{16M}$. The gap is at most 32, which is a dramatic improvement over the $\Theta(\log K)$ gap [1].

Without **Assumptions** 1-2, our new proposed scheme may perform much better than that in [1]. For example, let $\overline{T} = \Theta(\log K)$ and $N_l F_l^2 = N_1 F_1^2 / 2^l$ for $l \geq 2$. The achievable rate in [1] is $(\overline{T}+1)\sum_{l=1}^{\overline{T}} N_l F_l^2 / M = \Theta(\log K \cdot N_1 F_1^2 / M)$. The achievable rate in this paper is $(\sum_{l=1}^{\overline{T}} \sqrt{N_l} F_l)^2 / M = \Theta(N_1 F_1^2 / M)$, which is reduced by a factor of $\Theta(\log K)$. Thus, this new achievable scheme is also critical in improving the overall performance.

A. Relaxing Assumptions 1 & 2

When **Assumptions** 1-2 are relaxed, the analysis becomes much more complicated since there are many corner cases. Here we provide the main intuitions to relax **Assumptions** 1-2 in Prop. 1, and then match it to the achievable rate.

Thanks to **Observation** 1, we can remove files, add files with zero packets, or move files across types to generalize Prop. 1, as long as $|\mathbb{F}_l| = 2 \cdot 4^{l-2} N_1$, for $l \geq 2$. Further, we have two more observations below.

Observation 2: In **Assumption** 1, it requires that $N_l = 2 \cdot 4^{l-2} N_1$, for $l \geq 2$. When this relationship does not hold, i.e., $N_l \neq 2 \cdot 4^{l-2} N_1$, Proposition 1 still hold with slight modification. If $N_l < 2 \cdot 4^{l-2} N_1$ (or $N_l F_l^2 < \frac{1}{2} N_1 F_1^2$), we can add some virtual files with zero size into file set $\mathbb{F}_l$, to satisfy $|\mathbb{F}_l| = 2 \cdot 4^{l-2} N_1$. If $N_l > 2 \cdot 4^{l-2} N_1$, we can remove some files from $\mathbb{F}_l$ and do not use them in the request patterns. In this case, $H(\mathbb{F}_l)$ in Proposition 1 can be seen as $\min(N_l, 2 \cdot 4^{l-2} N_1) \cdot F_l$.

Observation 3: With **Observation** 1, we can move some files with size F_l to a file set $\mathbb{F}_i$ ($i < l$) in the constructed patterns to increase their impact on the lower bound. To see this, suppose that for some $l \geq 2$, $N_l = a_l 4^{l-2} N_1$ where $a_l < 2$. The impact of these N_l files with size F_l on the lower bound $\left(\sum_l \frac{1}{\sqrt{N_1}} 2^{1-l} H(\mathbb{F}_l)\right)^2 / (4M)$ is determined by the term $\frac{1}{\sqrt{N_1}} 2^{1-l} H(\mathbb{F}_l)$, which equals to $a_l \sqrt{N_1} F_1 / 4$. However, if we move these N_l files to the set $\mathbb{F}_i$ (where i satisfies $N_l = 2 \cdot 4^{i-2} N_1$), the impact on the lower bound is then $\frac{1}{\sqrt{N_1}} 2^{1-i} H(\mathbb{F}_i) = \frac{1}{\sqrt{N_1}} 2^{1-i} N_l F_l = \sqrt{\frac{a_l}{8}} \sqrt{N_1} F_1$, which could be much larger when a_l is small (e.g., $a_l = \frac{2}{l}$).

On the other hand, the impact of these N_l files on the achievable rate (17) is determined by $\sqrt{N_l} F_l$, which is $\sqrt{\frac{a_l}{4}} \sqrt{N_1} F_1$ and thus is in consistence with the term $\sqrt{\frac{a_l}{8}} \sqrt{N_1} F_1$ shown in **Observation** 3. The gap can then be bounded as a constant.

Note that the file movement in **Observation** 3 is more dependent on the number of files with a same type, but less dependent on the file sizes, which is consistent with the design of our achievable scheme.

Further note that in the above observations, we do not consider the integer constraints such as that in **Assumption** 2. To relax the integer constraints, one can employ the proof for Proposition 2 in [1] to have the following results.

Proposition 2: When the sum size of all files is no smaller than $8M$, the gap between the achievable rate and the lower bound is at most a constant.

When the sum size of all files is smaller than $8M$, the caching probability for some files may approach 1 and the approximation $\frac{F_l}{q_l}$ in the achievable rate may not work. This case requires further investigation.

V. Conclusion

In this paper, we study coded caching for files with distinct file sizes. We derive a novel lower bound via new techniques. We also propose a new achievable scheme that cache files of a type with probability inversely proportional to the number of such files. When the total file size is larger than eight times the individual cache size, the gap between the achievable rate and the lower bound is shown to be at most a constant.

Acknowledgment

This work was partially supported by NSF China grants (No. 61601287, No. 61702205), and NSF grants (CNS-1703014, CCF-1422997, ECCS-1407604, CCF-1618475, CCF-1816013).

References

[1] J. Zhang, X. Lin, C.C. Wang, and X. Wang, "Coded Caching for Files with Distinct File Sizes", in *Proc. IEEE ISIT*, Hong Kong, 2015.

[2] 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.

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

[4] N. Karamchandani, U. Niesen, M.A. Maddah-Ali and S. Diggavi, "Hierarchical Coded Caching", in *IEEE Trans. Inform. Theory*, vol. 62, no. 6, pp. 3212-3229, June 2016.

[5] J. Hachem, N. Karamchandani and S. Diggavi, "Coded Caching for Multi-level Popularity and Access", in *IEEE Trans. Inform. Theory*, vol. 63, no. 5, pp. 3108-3141, Mar. 2017.

[6] U. Niesen, and M.A. Maddah-Ali, "Coded Caching with Nonuniform Demands", in *IEEE Trans. Inform. Theory*, vol. 63, no. 2, pp. 1146-1158, Feb. 2017.

[7] M. Ji, A. Tulino, J. Llorca and G. Caire, "Order-Optimal Rate of Caching and Coded Multicasting With Random Demands", in *IEEE Trans. Inform. Theory*, vol. 63, no. 6, pp. 3923-3949, Apr. 2017.

[8] R. Pulikkoonattu, "On the Inequalities in Information Theory", Technical Report, EPFL, Jan. 2008.