Coded Caching under Arbitrary Popularity Distributions

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

***Abstract*—Caching plays an important role in reducing the backbone traffic when serving high-volume multimedia content. Recently, a new class of coded caching schemes have received significant interest because they can exploit coded multi-cast opportunities to further reduce backbone traffic. Without considering file popularity, prior works have characterized the fundamental performance limits of coded caching through a deterministic worst-case analysis. However, when heterogeneous file popularity is taken into account, there remain open questions regarding the fundamental limits of coded caching performance. In this work, for an arbitrary popularity distribution, we first derive a new information-theoretical lower bound on the expected transmission rate of any coded caching schemes. We then show that a simple coded-caching scheme attains an expected transmission rate that is at most a constant factor away from the lower bound. Unlike other existing studies, the constant factor that we derived is independent of the popularity distribution.**

I. Introduction

As the amount of Internet traffic continues to grow, video is expected to dominate 69% of the overall traffic [2], which will greatly stress the underlying communication infrastructure. Historically, caching has played a significant role in reducing the bandwidth requirement for serving video traffic. By placing contents closer to, or even at the end-users, the bandwidth requirement at the upstream links can be greatly reduced. Most of such studies of caching have focused on the case where uncoded video packets were stored and transmitted (see, e.g., [3–6] and references therein).

Recently, a new class of caching schemes, called coded caching [7–16], have gained significant interest because it can significantly reduce the upstream bandwidth requirement in systems with broadcast/multicast capabilities. Consider K users requesting contents from one server through a shared communication link with broadcast capability. Each user may request any one of the N files ($N > K$), but each user only has a storage with size $M < N$. In the worst case, each user may request a distinct file. With conventional (uncoded) caching scheme, it is easy to see that the worst-case transmission rate on the upstream link must be $K(1 - \frac{M}{N})$, because each user can only cache $\frac{M}{N}$ fraction of all the contents. [7] refers to this factor $(1 - \frac{M}{N})$ as the *local* caching gain. Unless M is large (compared to N), this local caching gain will not differ significantly from 1 (i.e., the baseline with no-caching). Note that the broadcast capability of the system is not exploited here because each user can request a different file. In contrast, with the coded caching scheme in [7], the worst-case transmission rate at the upstream link is reduced to $K(1-\frac{M}{N})\frac{1}{1+KM/N}$. The additional factor $\frac{1}{1+KM/N}$, which is referred to as the *global caching gain* in [7], suggests a significant improvement over the uncoded case when the *global* storage capability KM of all users is comparable to, or larger than, N. The key idea of [7] is to transmit *coded* packets so that multiple users can benefit from the same broadcast packet. Thus, the broadcast capability in the system can be exploited even if different users request different files. [7] further shows in an information-theoretic sense that the worst-case transmission rate of the coded caching scheme in [7] is at most a constant factor (specifically, 12 times) away from the minimum possible. In this sense, the performance of the coded caching scheme of [7] is close to the fundamental limit for the system studied. The works in [8, 14–16] extend this idea to decentralized caching, hierarchical networks, multiple group-cast, and online caching, respectively.

The studies cited above all focus on the *deterministic* worst-case, i.e., not only does each user request distinct files, the performance of the system is studied against the worst-case request pattern. Arguably, if the popularity of the files are identical, the probability of each request pattern will vary less significantly. Then, the worst-case performance may not differ significantly from the average-case performance [9]. In reality, however, the file popularity *can* differ significantly, and thus some request patterns will occur much more frequently than other request patterns. As a result, the average-case performance can differ significantly from the worst-case bound (see also the discussions at the end of Section II).

While the average-case performance of coded caching under heterogeneous file popularity was studied in [9–13], the optimality bounds obtained are substantially weaker than the results in [7] because the gap between the achievable bound and the lower bound depends on various system parameters. Specifically, in [9], contents are divided into groups with similar popularity. Each group is assigned a separate portion of the cache and uses the coded caching scheme of [8]. The gap between the corresponding transmission rate and the lower bound is found to increase with the total number of groups.

An earlier version of this paper has appeared in Information Theory and Applications Workshop, UCSD, Feb. 2015 [1].

[10] studies a popularity model that can be viewed as an intermediate between worst-case analysis and average-case analysis. It models non-uniform popularity by assuming that the file popularity has L different levels. Both the number of files at each level and the number of users requesting files at each level are fixed. The authors then study the deterministic worst-case performance under such a setting. When the number of users is large, this model can be seen as an approximation of a stochastic-demand model. The theoretical gap between the achievable transmission rate and the lower bound established in [10] increases as L^3. The work in [13] is most related to ours. In [13], the authors propose a large class of achievable schemes, called RAP-GCC and RAP-CIC, which place files in caches according to a popularity-dependent caching distribution. However, because the optimal caching distribution for RAP-GCC and RAP-CIC may not have analytically tractable solutions in general, [13] then focuses on a sub-class of RAP-GCC, called RLFU (Random Least-Frequently-Used caching), to study the performance gap between the lower bound and achievable bound of the required transmission rate. RLFU uses a clever and surprisingly simple caching distribution that divides the contents into 2 groups. Although both the lower bound and the achievable bound (by RLFU) are given in [13] for arbitrary popularity distributions, the gap between the achievable bound and the lower bound is shown to be a constant only when the file popularity follows a Zipf distribution. Note that the gaps estimated by the theoretical results of [13] depend on the parameters of the Zipf distribution, and may also become large for certain ranges of the parameter values [13]. Further, the constant factors are only shown in the asymptotic limit when the number of files and/or the number of users are large. Therefore, it remains an important open question what is the fundamental limit of the performance of coded caching for the more practical scenario of heterogeneous file popularity, and whether one can find a coded-caching scheme whose performance gap from the lower bound is *independent* of the popularity distribution even in the non-asymptotic settings.

In this paper, we make the following contributions to answer the above open question. First, we show that a simple coded-caching scheme (which is a special case of RAP-GCC in [13] and is also similar to RLFU) can attain an average transmission rate that is at most 55 times from the optimal[1]. Although this factor appears to be large, it is the first result in the literature with a constant-factor gap that is independent of the popularity distributions and the system size. In contrast, as we discussed earlier, the performance gap in prior studies could be arbitrarily large depending on either the number of groups [9], the number of levels [10], or the parameter of the Zipf distribution [11–13]. Second, a key step towards this result is to use a new construction in Section IV to establish a sharper lower bound (in the order sense) on the achievable transmission rate of *any* schemes. Specifically, our lower bound may be higher than the lower bound in [13] by an arbitrarily large factor under certain popularity distributions (see further discussions in Section III). We establish this lower bound by a series of *reduction* and *merging* steps that convert the original system with heterogeneous popularity to other systems with uniform popularity. Using these techniques, we are able to quantify the impact of both "popular" files and "non-popular" files, which we believe is the main reason that we can obtain sharp constant-factor characterizations even in the non-asymptotic settings. (See further discussions at the end of Section IV.D.) These techniques may be of independent interest for future studies of coded caching performance. Third, while our achievable scheme is similar to RLFU proposed [13] (in the sense that for most parameter settings we divide the files into 2 groups, correspondingly to popular and unpopular files, respectively), there are parameter settings where we need to divide the files into 3 groups. This new modification turns out to be important for attaining the constant-factor performance gap under *all* popularity distributions and non-asymptotic settings. Further, our division between popular and unpopular files uses a simple threshold N_1. Specifically, suppose that the number of users is K and the size of each user's storage is M. Then, the threshold N_1 corresponds to the least-popular file among those whose popularity is no smaller than $\frac{1}{KM_r}$, where $M_r = \max(M, 3)$. Compared with the choice of the threshold (denoted by $\tilde{m}$) in [13] (either chosen as a function of the Zipf distribution in the theoretical analysis, or obtained via a 1-dimensional optimization over an achievable bound), our choice of N_1 is in a simple closed-form that works for arbitrary popular distributions (see further discussions in Section III). Finally, note that the RLFU scheme in [13] with the optimal threshold $\tilde{m}$ should in general attain a better performance than the simpler choice of N_1 in this paper. Thus, it implies that RLFU combined with appropriately dividing the files into 3 groups under certain settings also attains an average transmission rate that is away from the optimal by at most a constant factor, independently of the popularity distribution. (This immediately follows that the best scheme in the larger class of RAP-GCC will also attain a constant-factor gap.) However, the RLFU scheme itself without the 3-group modification does not have this constant-factor performance guarantees under arbitrary popularity distribution (see Section III).

The remainder of this paper is as follows. We first present the network model in Section II. Main results are summarized in Section III. Followed is the analysis on the information theoretical lower bound in Section IV. The achievable scheme is analyzed in Section V. The gap between the lower bound and the achievable rate is derived in Section VI. Simulation results are presented in Section VII. Then, we conclude.

II. Network Model

In the following, we present the network model for a video delivery system with both local caches and broadcast capabilities (see Figure 1).

We assume that there are N distinct files from the set $\mathcal{F} = \{F_1, F_2, ..., F_N\}$. The popularity of the file F_i is p_i, where $\sum_{i=1}^{N} p_i = 1$. Without loss of generality, we assume that the file size is of unit length and the file popularity is decreasing in the index, i.e., $p_i \geq p_j$ if $i \leq j$.

[1] Note that the conference version of this paper [1] not only shows a larger multiplicative factor, but also requires a small additive factor. The results in this paper use a more-refined analysis that eliminates the need for the additive factor.

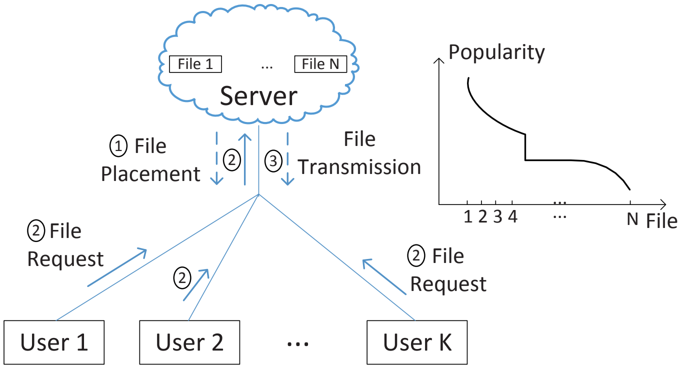


Fig. 1: An illustration of the network model.

There is one server who has all N files and who serves these files to K users interested in these files. Each user has a local cache with size M (again measured with respect to the unit-length of the files). The K users are connected to the server through a network with broadcast capability, i.e., each transmission from the server can be received by all users.

Before users request any files, some of the contents are placed in the users' caches. This is called cache placement and in practice is usually carried out during off-peak hours of the network. Then, at each time, a user k will request file F_i with probability p_i, independently of all other users and files. If the user's local cache already has (some of) the content, the request can be served locally. Otherwise, the server must transmit (via broadcast) contents not available from the local cache. The goal is that every user should be able to reconstruct the file that it requests with the information received from the server and the cached content in its local cache.

A. Definition of the Expected Transmission Rate

In this subsection, we will define the expected rate needed from the server in serving the requests. Note that we do not consider the transmission rate for cache placement.

Let $W_i = \{f_{i1}, f_{i2}, ..., f_{iK}\}$ denote a request pattern, where $f_{ij} \in \mathcal{F}$ is the requested file for the j-th user, $1 \leq j \leq K$. Note that there are N^K such patterns. Let $\mathbb{W}$ be the set of all possible request patterns from K users, i.e., $\mathbb{W} = \{W_1, W_2, ..., W_{N^K}\}$. Since each user can request one file from N files independently, the probability for event W_i is given by

$$P(W_i) = \prod_{j=1}^{K} \mathcal{P}(f_{ij})$$

where $\mathcal{P}(f_{ij})$ is the probability for a user to request file f_{ij}. Note that in our original system, the probability for a user to request file F_j is p_j. However, later in the analysis we will compare to another system with a different popularity distribution $\mathcal{P}$. Hence, we use the notation $\mathbb{W}(K, \mathcal{F}, \mathcal{P})$ to denote the set $\mathbb{W}$ of possible request patterns associated with the corresponding popularity distribution $\mathcal{P}$.

Obviously, given a set of files $\mathcal{F}$ and the files' corresponding popularity distribution $\mathcal{P}$, there exists numerous caching and transmission schemes to meet users' request. For a caching and transmission scheme $\mathfrak{F}$, let $r_{\mathfrak{F}}(K, W_i)$ denote the amount of broadcast transmission from the server that is needed to satisfy a request W_i. The expected rate under scheme $\mathfrak{F}$ is therefore defined as

$$R_{\mathfrak{F}}(K, \mathcal{F}, \mathcal{P}) = \sum_{i=1}^{N^K} r_{\mathfrak{F}}(K, W_i) P(W_i). \tag{1}$$

We wish to find the schedule $\mathfrak{F}$ that minimizes $R_{\mathfrak{F}}(K, \mathcal{F}, \mathcal{P})$. Define the optimal rate as

$$R(K, \mathcal{F}, \mathcal{P}) = \min_{\mathfrak{F}} R_{\mathfrak{F}}(K, \mathcal{F}, \mathcal{P}). \tag{2}$$

Unfortunately, finding the exact optimal schedule that achieves this optimal rate is very difficult [7–16]. Like [7–16], our goal is to find a simple scheme $\mathfrak{F}$ whose achievable rate is as close to the optimal rate $R(K, \mathcal{F}, \mathcal{P})$ as possible.

Remark: In [7], instead of studying the expected rate (1), the authors focus on the worst-case rate, i.e.,

$$\max_{W_i} r_{\mathfrak{F}}(K, W_i). \tag{3}$$

Let $\mathfrak{F}'$ be the optimal scheme that attains the minimum value of (3), and let $\mathfrak{F}$ be the scheme proposed in [7]. Then [7] shows that

$$\frac{\max_{W_i} r_{\mathfrak{F}}(K, W_i)}{\max_{W_i'} r_{\mathfrak{F}'}(K, W_i')} \leq 12. \tag{4}$$

However, in this paper since we are interested in the expected rate given in (2), we would be interested in the gap

$$\frac{\sum_{i=1}^{N^K} r_{\mathfrak{F}}(K, W_i) P(W_i)}{\sum_{i=1}^{N^K} r_{\mathfrak{F}^*}(K, W_i') P(W_i')}, \tag{5}$$

where $\mathfrak{F}^*$ is the optimal scheme that attains the minimum value of $\sum_{i=1}^{N^K} r_{\mathfrak{F}}(K, W_i') P(W_i')$. Note that the bound in (4) does not imply that the expression in (5) is bounded by the same constant, especially when the probability $P(W_i)$ varies significantly. In general, even if the bound in (4) holds, the expression in (5) can still be arbitrarily large. Thus, quantifying the performance gap in terms of the expected rate represents a new research problem.

III. Main Results

In this section, we provide an overview of our main results. Given an arbitrary popularity distribution, our first result establishes a fundamental lower bound on the expected transmission rate for any coded caching scheme. Let $[x]_+$ denote $\max(0, x)$. Recall that the files are indexed with non-increasing popularity. Without loss of generality, we assume $p_{N+1} = 0$ (which ensures the existence of N_1 defined next even when $p_N \geq \frac{1}{KM_r}$). Let $M_r = \max(M, 3)$, and let N_1 be the integer that satisfies $p_{N_1} \geq \frac{1}{KM_r}$ and $p_{N_1+1} < \frac{1}{KM_r}$. In other words, N_1 is the least popular file among those whose popularity is no smaller than $1/KM_r$. If no such files exist, let N_1=0. Let $N_2 = \left\lfloor \frac{\sum_{i>N_1} p_i}{2/KM_r} + \frac{1}{2} \right\rfloor$. Further, for any integer N_x between 1 and $N-1$, let $N_y(N_x) = \left\lfloor \frac{\sum_{i>N_x} p_i}{2p_{N_x}} + \frac{1}{2} \right\rfloor$. Then, we have the following.

Theorem 1: With K users requesting files independently in $\mathcal{F}$ according to the corresponding popularity distribution $\mathcal{P}$, the lower bound on the expected transmission rate is given by

$$R(K,\mathcal{F},\mathcal{P}) \geq \frac{1}{11}\max\Big\{\frac{1}{M_r}(N_1+N_2-M), \max_{N_x\geq N_1+1} Kp_{N_x}[N_x+N_y(N_x)-M]\Big\}. \tag{6}$$

The new lower bound in Theorem 1 is one of the main contributions of the paper. It is sharper than those reported in the literature [9, 11–13] in the order sense. Specifically, we will show shortly that our lower bound can be higher than the lower bound in [13] by an arbitrarily large factor under certain popularity distributions. Thus, this sharper bound is the main reason behind the improved performance characterization reported in this paper. We acknowledge that there will be settings where our lower bound can be lower than that of [13]. However, even when this happens, the difference is at most a constant multiplicative factor since Theorem 2 establishes an achievable upper bound that is at most a constant factor away from our lower bound. Here, the index N_1 in the first term of (6) plays an important role in most of the results in this paper. Recall again that the popularity p_i is non-increasing in the file index i. Roughly speaking, N_1 is the index for the file whose popularity is around $\frac{1}{KM_r}$. We may view all files $i \leq N_1$ as the "more popular" files, and all files $i > N_1$ as the "unpopular" files. As readers will see in the proofs of Theorem 1 in Section IV, we will first show that the expression $\frac{1}{11M_r}[N_1-M]_+$ is a lower bound on the expected transmission rate for serving the more popular files. Existing results in [9, 11–13] use different variations of this expression as the lower bound. However, since this expression neglects the contribution of the unpopular files, it results into poorer performance characterization. In contrast, we show that the contribution of the unpopular files can be accounted for by having N_2 in the first term of (6). As readers will see in Section IV.D, here we introduce a novel "merging" procedure that merges multiple unpopular files into one file with the sum of the popularities, so that the new popularity is greater than or equal to $1/(KM_r)$. In this way, N_2 can be interpreted (approximately) as a lower bound on the number of such "merged" files. Thus, N_1 and N_2 combined produce the lower bound in the first term of Theorem 1. However, the first term of (6) would be trivial if $N_1 + N_2 \leq M$. In that case, we will need to increase the threshold index N_1 to a larger value N_x, in order to obtain a sharper lower bound. The second term in (6) takes care of this case. Details of the proof will be presented in Section IV.

Difference from [13]: As we discussed above, a key difference between our lower bound and the prior results is that our lower bound accounts for the contribution of the unpopular files. As one can see from the example below, accounting for the contribution of the unpopular files to the lower bound is critical, because otherwise the lower bound may be looser by an arbitrarily factor. Take Zipf distribution with $\alpha = 1$ as an example. Suppose that $M \geq 3$ and the number of files N is large. Then, using the approximation that $\sum_{i=1}^{N}\frac{1}{i} \approx \log N$, it is easy to see that $p_i \approx \frac{1}{i\log N}$, and thus the threshold N_1 satisfies $N_1 \approx \frac{KM}{\log N}$. The lower bound due to these N_1 popular files is $\frac{1}{11}(\frac{N_1}{M}-1) \approx \frac{1}{11}(\frac{K}{\log N}-1)$ (assuming $M \geq 3$). Thus, this lower bound is always less than $\frac{1}{11}\frac{K}{\log N}$. Note that using N_x in the place of N_1 will not get a lower bound higher than $\frac{1}{11}\frac{K}{\log N}$ either. To see this, if we do not consider the term $N_y(N_x)$, the lower bound in the second term of Theorem 1 gives $\frac{1}{11}Kp_{N_x}(N_x-M)$. Since $Kp_{N_x}N_x \approx \frac{K}{\log N}$, this lower bound can not be larger than $\frac{1}{11}\frac{K}{\log N}$ either. On the other hand, the lower bound given by Theorem 1 can be arbitrarily larger once the contribution for unpopular files is accounted. For example, when $N_1 \leq KM$, the impact from unpopular files, $\frac{N_2}{M_r}$, can be approximated as $\sum_{i>N_1}Kp_i = K - K\sum_{i\leq N_1}Kp_i \approx K - \frac{K\log N_1}{\log N}$. When $\log N \gg \log KM$, we would have $\frac{N_2}{M_r} \approx O(K)$. In other words, accounting for the impact of unpopular files can increase the lower bound by a factor as large as $O(\log N)$. Due to this reason, it is critical to use the improved lower bound in Theorem 1 in order to prove constant-factor performance gaps under arbitrary popularity distributions.

We next present an achievable scheme that can attain a corresponding upper bound. Our achievable scheme is a special case of RAP-GCC in [13] and is also similar to RLFU. Recall that each file is of unit length. In order to allow a portion of each file to be cached, we refer to a minimally divisible portion of a file as a "bit", and assume that each file has $|F|$ "bits". We are most interested in the case of large files, i.e., when the bits are very small compared to the file size, and hence $|F| \rightarrow +\infty$. Our achievable scheme is similar to RLFU in most settings (i.e., when $N_1 \geq M$ and $M \geq 3$), but different in other settings (i.e., when $N_1 < M$ or $M < 3$). Specifically, when $N_1 \geq M$ and $M \geq 3$, our proposed achievable scheme uses the decentralized coded caching scheme of [8] to serve the "popular" files, and uses uncoded transmissions to serve the "unpopular" files. Each user randomly caches an equal number of $\frac{M|F|}{N_1}$ bits from every file $F_1, ..., F_{N_1}$. The remaining unpopular files are *not* cached. After the users request the files according to the popularity distribution $p_1, ..., p_N$, the decentralized *coded* transmission scheme of [8] is used to serve those users requesting popular files, and an *uncoded* transmission scheme is used to serve those users requesting unpopular files. We note that this part of the scheme is similar to the Random LFU scheme studied in [13]. However, when $N_1 < M$ or $M < 3$, we do not use RLFU. Instead, we divide the files into 3 groups and serve them separately. Specifically, we cache all files 1 to $\lfloor M \rfloor$, cache $M - \lfloor M \rfloor$ portion of file $\lfloor M \rfloor + 1$, and not to cache all other files. Note that this scheme can be seen as LFU with fractional caching of the file $\lfloor M \rfloor + 1$ (when M is not an integer). The following result summarizes an upper bound on the expected transmission rate of our achievable scheme.

Theorem 2: With K users independently requesting files in $\mathcal{F}$ according to the popularity distribution $\mathcal{P}$, as $|F| \rightarrow +\infty$, the optimal achievable rate can be upper bounded by

$$R(K,\mathcal{F},\mathcal{P}) \leq \min\Big(\frac{[N_1-M]_+}{\max(1,M)} + \sum_{i>N_1}Kp_i, Kp_{N_3}(N_3-M) + \sum_{i>N_3}Kp_i\Big), \tag{7}$$

TABLE I: Comparisons of our work and [13]

	Our work	[13]
System settings & Performance gap	Performance gap (between achievable schemes and lower bounds) is derived for arbitrary distribution and system size. Constant-factor gap independent of the popularity distribution and system size.	Performance gap (between achievable schemes and lower bounds) is derived only for Zipf distribution and for asymptotic settings when the system size is large. Gap depends on the parameter α of the Zipf distribution, and may approach infinity as α approaches 1.
Lower bound	Lower bound accounts for the contribution of both popular files and unpopular files.	Lower bound only accounts for the l most-popular files, and thus may be arbitrarily lower than our lower bound for certain popularity distributions.
Achievable bound	The achievable scheme divides all files into either 2 groups or 3 groups depending on system settings. Using a simple expression for N_1, this scheme is shown to be constant-factor away from the lower bound under arbitrary popularity distribution.	The achievable bound is analyzed based on RLFU, which divides the files into 2 groups. As a result, RLFU may incur an arbitrarily large gap from the lower bound under certain popularity distributions.

where $N_3 = \lfloor M \rfloor + 1$.

In the first term of (7), when $N_1 \geq M$ and $M \geq 3$, $\frac{[N_1-M]_+}{\max(1,M)} = \frac{[N_1-M]_+}{M}$ is an upper bound on the expected transmission rate to serve the more popular files (i.e., with index $i \leq N_1$), and $\sum_{i>N_1} Kp_i$ is an upper bound on the expected transmission rate to serve the unpopular files. However, the expression $\frac{[N_1-M]_+}{M} + \sum_{i>N_1} Kp_i$ may become too loose in certain cases (e.g, when $N_1 < M$ or $M < 3$). In those cases, coded caching has little gain, and thus we use uncoded transmissions for all files, which leads to both the second term in (7) and the term $\frac{[N_1-M]_+}{\max(1,M)}$. Consider the scenario when $N_1 \geq M$ and $M \geq 3$, and thus the first term of (7) dominates. Note that increasing N_1 by 1 will increase $\left[\frac{N_1}{M} - 1\right]_+$ by $1/M$, and will reduce $\sum_{i>N_1} Kp_i$ by roughly Kp_{N_1}. Thus, by setting $p_{N_1} \approx \frac{1}{KM}$, this index N_1 is chosen such that the net effect to the first term of upper bound (7) is approximately zero, and thus the first term in (7) is approximately minimized. This property was the main intuition behind our choice of N_1. However, the analysis in the paper will account for all scenarios (not just $N_1 \geq M$ and $M \geq 3$).

Difference from [13]: Note that the above modification that divides the files into 3 groups in some settings turns out to be crucial for attaining the constant-factor performance gap under arbitrary popularity distributions, especially when the optimal transmission rate is small. The difference from RLFU occurs when $N_1 < M$ and when M is not an integer, i.e., when the cache can hold all popular files and the cache size is not a multiple of the file size. In this case, our optimal scheme is to cache all files 1 to $\lfloor M \rfloor$, cache $M - \lfloor M \rfloor$ portion of file $\lfloor M \rfloor + 1$, and not to cache all other files. To compare the performance difference with RLFU: assuming that there are $\lfloor M \rfloor + 1$ files. Let $N_3 = \lfloor M \rfloor + 1$. If RLFU caches at most $\lfloor M \rfloor$ files, the average transmission rate is $R_1 = \Theta(Kp_{N_3})$ when $Kp_{N_3} = O(1)$, which is simply the rate for RLFU to serve file N_3 in an uncoded manner when any users request it. On the other hand, if RLFU caches all $N_3 = \lfloor M \rfloor + 1$ files, the average transmission rate can be lower bounded by $R_2 = 1 - \frac{M}{N_3} = \frac{\lfloor M \rfloor + 1 - M}{\lfloor M \rfloor + 1}$, which corresponds to the rate when there is only one file being requested. In contrast, by caching all files 1 to $\lfloor M \rfloor$ and partially caching file N_3, our rate is $R_3 = Kp_{N_3}(\lfloor M \rfloor + 1 - M)$. We note that, depending on the system setting, R_3 can be an arbitrarily small fraction of $\min\{R_1, R_2\}$. For example, for any integer H, by letting $N = H$, $M = H - \frac{1}{H}$, and $p_i = \frac{1}{H-1} - \frac{1}{KH^2(H-1)}$ (i=1,...,$H-$1), $p_H = \frac{1}{KH^2}$, we have $R_1 = \Theta(\frac{1}{H^2})$, $R_2 = \Theta(\frac{1}{H^2})$ and $R_3 = \Theta(\frac{1}{H^3})$. Thus, as H goes to infinity, the ratio $\frac{R_3}{\min\{R_1,R_2\}}$ goes to zero. Since our achievable scheme is order optimal, it implies that directly using RLFU is not order-optimal for this case. Hence, this part of dividing files into 3 groups (one group is cached in its entirety, one file is cached partially, other files are not cached) turns out to be critical for the constant-gap result. We acknowledge that the difference may be small under Zipf distribution. However, since we are focusing on arbitrary distribution and non-asymptotic case in this paper, capturing this subtle difference is critical for the overall results.

From Theorem 1 and Theorem 2, we will show in Section VI that the gap between the lower bound R_{lb} and the upper bound R_{up} is bounded by a multiplicative constant.

Corollary 1: With K users independently requesting files in $\mathcal{F}$ according to the popularity distribution $\mathcal{P}$, as $|F| \to +\infty$, the lower bound R_{lb} and the upper bound R_{up} can be bounded by

$$R_{up} \leq 55 R_{lb}. \tag{8}$$

Thus, the bounds differ at most by a multiplicative factor of 55. As we discuss in the introduction, although this factor may appear to be large, it is the first result in the literature with a constant-factor gap that is independent of the popularity distributions. In contrast, the gap (between upper- and lower-bounds) estimated by the existing results can be arbitrarily large depending on either the number of groups [9], the number of levels [10], or the parameter of the Zipf distribution [13]. It is remarkable that such a simple coded caching scheme, with a very simple choice of N_1, can achieve such a strong performance guarantee, independently of the popularity distribution.

As we mentioned earlier, although both our lower bound and achievable bound have some similarity to those in [13], our lower bound is sharper in the order sense because it accounts for the contribution of unpopular files, and our achievable bound is also tighter in the order sense because it divides the files into 3 groups under some parameter settings. We believe that such difference is the main reason why we can attain constant-factor performance guarantees independent of the popularity distribution, while the performance gap in [13] only holds when the file popularity follows a Zipf distribution and when the system sizes approach infinity, and even then the performance gap in [13] still depends on the parameter α of Zipf distribution (which may approach infinity as α approaches 1). A brief comparison with [13] is presented in Table I.

We also note that, for certain ranges of the exponent α of the Zipf distributions, the performance characterization in [13] for RLFU may be tighter than the 55 factor reported in (8). Further, both our achievable scheme and RLFU are special cases of the larger class of RAP-GCC schemes reported in [13]. Thus, the results in [13] and in this paper combined provide a more complete characterization of the performance guarantees for coded caching schemes across both Zipf and non-Zipf distributions.

A. *Main Intuition*

Before we present the proofs for these main results, we would like to illustrate the main intuition behind. First, consider only the "popular files" 1 to N_1, i.e., assuming that the unpopular files N_1+1 to N are removed. Let us refer to this system as "System 1". In our proof, we will consider an alternate system where the popularity of all popular files is reduced to $\frac{1}{KM_r}$. We will refer to this alternate system as "System 2" (see Section IV-A). Intuitively, the average transmission rate in System 2 is no larger than that in System 1. Further, since all files are with the same popularity in System 2, the average-case and the worst-case performance will not differ too much [9]. Thus, one can then use System 2 to derive a lower bound on the average transmission rate, and compare it with an upper bound attained by an achievable scheme.

However, the potential problem of this argument is that, when we reduce the popularity of all popular files to $\frac{1}{KM_r}$, some popularity values could be reduced by several orders of magnitude. It is then unclear why the lower bound derived from System 2 is still a reasonable lower bound for System 1. The intuition behind this insensitivity can be explained as follows. Suppose that there are K' users in System 1 that request any of the popular files. Then, according to the result in [7], the worst-case transmission rate to serve these K' users is no larger than

$$K'(1-\frac{M}{N_1})\frac{1}{1+\frac{K'M}{N_1}}. \tag{9}$$

Now, suppose that the individual cache size M is much smaller than N_1, and the global cache size $K'M$ is much larger than N_1 (note that this is precisely the regime where coded caching will be most helpful [7]). Then, we have $1-\frac{M}{N_1}\approx 1$ and $1+\frac{K'M}{N_1}\approx\frac{K'M}{N_1}$. Thus, the expression in (9) is approximately equal to N_1/M. The significance of this observation is that this approximated expression is independent of K'. In other words, in a suitable regime of interest, the exact popularity of the "popular files" does not seem to matter! It is then plausible to argue that, even when we reduce the popularity values to $\frac{1}{KM_r}$ in System 2, there is no substantial change in the lower-bound performance. Of course, this argument needs to be carefully made. Further, we have to account for not only popular files, but also unpopular files. The proofs in the next section will make this intuition precise.

IV. Lower Bound on the Expected Rate

In this section, we present the proof of Theorem 1, i.e., the lower bound.

The proof consists of two parts. Subsections A-C focus on popular files 1 to N_1, and prove the part that $R(K,\mathcal{F},\mathcal{P})\geq\frac{1}{11M_r}(N_1-M)$, where $M_r=\max\{M,3\}$. This proof is composed of 5 steps. From the first step to the fourth one, we map the original system into a series of reduced systems, whose information-theoretical rate is strictly smaller than previous ones. Then, we bound the rate needed for popular files in the fourth step. We note that the ideas used in the 1st to 4th reduction steps are similar to [9][13]. In the fifth step, we introduce a novel "merging" technique in Subsection D to account for the unpopular files and prove the part that $R(K,\mathcal{F},\mathcal{P})\geq\frac{1}{11M_r}(N_1+N_2-M)$. Finally, in Subsection E, we deal with the case when $N_1+N_2\leq M$, and establish the second term in (6). A brief summary of the constructed systems are presented in Table II. As we discussed earlier, while some of the techniques for quantifying the impact of popular files are similar to [9][13], our treatment of unpopular files is new and is the key reason for the sharper constant-factor characterization in our paper.

A. *Reduction Steps 1 & 2*

Recall that the set of files is given by $\mathcal{F}=\{F_1,F_2,...,F_N\}$ and their popularity distribution is given by $\mathcal{P}=\{p_1,p_2,...,p_N\}$. Next, we will compare to a series of reduced systems with different sets of files and popularity distributions. Again, let N_1 be the integer defined in Theorem 1.

In the first constructed system, the set of files is given by $\mathcal{F}_1=\{F_0,F_1,F_2,...,F_{N_1}\}$, where F_0 denotes the empty file, which is introduced for ease of presentation. Its corresponding popularity distribution is $\mathcal{P}_1=\{p_0,p_1,p_2,...,p_{N_1}\}$ and $p_0=1-\sum_{i=1}^{N_1}p_i$. In other words, we replace all unpopular files $F_{N_1+1},...,F_N$ in the original system by the empty file F_0, and reassign their popularity all to F_0. Intuitively, the

TABLE II: Brief introduction of constructed systems

Systems	Definition
System 1	Map the "unpopular" files from N_1+1 to N to a virtual empty file with their sum probability.
System 2	Reduce the popularity of files F_1,..., F_{N_1} to $\frac{1}{KM_r}$. Some users may request a same file.
System 3	With probability $\pi(K_1)$, *all* users request the empty file. With probability $1-\pi(K_1)$, exactly K_3 users request K_3 distinct files, and other users request empty files.
System 4	In each request pattern, exactly K_3 users request K_3 distinct files, and other users request empty files.
$(K,\mathcal{F}_3,\mathcal{P}_3)$	Reduce the the popularity of files F_1,..., F_{N_1} in $(K,\mathcal{F},\mathcal{P})$ to $\frac{1}{KM_r}$.
$(K,\mathcal{F}_4,\mathcal{P}_4)$	Merge the unpopular files into virtual files with popularity $\geq \frac{1}{KM_r}$ and $< \frac{2}{KM_r}$.
$(K,\mathcal{F}_4,\mathcal{P}_5)$	Reduce the probability of all files (except the empty file) to $\frac{1}{KM_r}$, which is similar to the System 2.

new system should require a lower transmission rate than the original system, which is stated in the following lemma.

Lemma 1: Let $R(K,\mathcal{F}_1,\mathcal{P}_1)$ be the minimum expected rate required to meet the requests by the K users, each of which randomly requests a file in $\mathcal{F}_1$ according to the popularity distribution $\mathcal{P}_1$. We have

$$R(K,\mathcal{F},\mathcal{P}) \geq R(K,\mathcal{F}_1,\mathcal{P}_1). \tag{10}$$

Proof: For a given cache placement and transmission scheme $\mathfrak{F}$, let $r = r_{\mathfrak{F}}(W_i)$ be the transmission rate for the random request pattern W_i in $\mathbb{W}(K,\mathcal{F},\mathcal{P})$, where W_i is chosen randomly in $\mathbb{W}(K,\mathcal{F},\mathcal{P})$. Note that since W_i is random, r is also a random variable. Further, the average transmission rate is $R_{\mathfrak{F}}(K,\mathcal{F},\mathcal{P}) = E[r]$. Similarly, let $r' = r_{\mathfrak{F}}(W_j)$ denote the transmission rate, where W_j is chosen randomly from the new distribution in $\mathbb{W}(K,\mathcal{F}_1,\mathcal{P}_1)$. Again, $R_{\mathfrak{F}}(K,\mathcal{F}_1,\mathcal{P}_1) = E[r']$. Note that the two expectations corresponding to two distributions, driven by $\mathbb{W}(K,\mathcal{F},\mathcal{P})$ and $\mathbb{W}(K,\mathcal{F}_1,\mathcal{P}_1)$, respectively. Therefore, r and r' cannot be directly compared in a pointwise manner, e.g., the number of request patterns in $\mathbb{W}(K,\mathcal{F},\mathcal{P})$ is larger than that in $\mathbb{W}(K,\mathcal{F}_1,\mathcal{P}_1)$.

In the following, we will show that $r' \leq^D r$, i.e., r' is stochastic dominated by r. To do so, we will use Theorem 3.1 in [19]. Specifically, we need to find two coupled variables $\hat{r}$ and $\hat{r'}$ with the following properties:

(a) r and $\hat{r}$ have the same distribution;

(b) r' and $\hat{r'}$ have the same distribution;

(c) $\hat{r} \geq \hat{r'}$ almost surely.

Then, applying Theorem 3.1 in [19], we can conclude that $r' \leq^D r$ and $E(r) = E(\hat{r}) \geq E(\hat{r'}) = E(r')$.

It remains to show the existence of $\hat{r}$ and $\hat{r'}$, which are constructed as follows. For every $W_i \in \mathbb{W}(K,\mathcal{F},\mathcal{P})$, we let $\hat{r} = r$ and thus they must have exactly the same distribution, satisfying property (a). We then map W_i to another request W'_i, such that the first N_1 elements in W_i remain the same as in W'_i, while other elements are mapped to the empty file. Let $\hat{r'}$ be the transmission rate to serve the request pattern W'_i.

By such a construction, we now verify property (b) holds. We wish to show that the probability with which the mapped request patterns is $W'_i = \{f_1, f_2, ..., f_K\}$, where f_k is the file requested by user k, is the same as the probability with which the same pattern $W_j = \{f_1, f_2, ..., f_K\}$ is requested in $\mathbb{W}(K,\mathcal{F}_1,\mathcal{P}_1)$. We first fix a user k and compare the probability that user k requests file f_k. If $f_k \in \{F_1, F_2, ..., F_{N_1}\}$, since we did not change their popularity, the probability $p_{W'_i}(f_k)$ that user k requests f_k in the mapped request pattern is the same as the probability $p_{W_j}(f_k)$ that user k requests f_k in $\mathbb{W}(K,\mathcal{F}_1,\mathcal{P}_1)$. If $f_k = \emptyset$, according to our construction, the probability $p_{W'_i}(f_k)$ that user k requests f_k in the mapped request pattern is equal to the probability that user k requests any file in $\{F_{N_1+1}, ..., F_N\}$ in $\mathbb{W}(K,\mathcal{F},\mathcal{P})$, which is $\sum_{i=N_1+1}^{N} p_i = p_0$. Hence, it is also the same as the probability $p_{W_j}(f_k)$ that user k requests the empty file in $\mathbb{W}(K,\mathcal{F}_1,\mathcal{P}_1)$. Finally, the probability that the mapped request pattern is W'_i, is

$$\begin{aligned} P(W'_i = \{f_1, ..., f_K\}) &= \prod_{k=1}^{K} p_{W'_i}(f_k) \\ &= \prod_{k=1}^{K} p_{W_j}(f_k) \\ &= P(W_j = \{f_1, ..., f_K\}). \end{aligned}$$

Thus, W'_i has the same distribution as W_j, and hence $\hat{r'}$ and r' have the same distribution, satisfying property (b).

Further, for any caching and transmission scheme that can satisfy users' request W_i, it must can satisfy W'_i, since W'_i can be seen as a subset of W_i. Hence, the rate to serve the request pattern W'_i is clearly no larger than the rate to serve the request pattern W_i. Thus we have $\hat{r} \geq \hat{r'}$ almost surely, satisfying property (c). The result of the lemma then follows. ■

We next create another new system by a further adjustment on the tuple $(K,\mathcal{F}_1,\mathcal{P}_1)$. Note that $N_1 \leq KM_r$. Otherwise, we will have $\sum_{i=1}^{N} p_i > KM_r \cdot p_{N_1} \geq 1$, which is a contradiction to $\sum_{i=1}^{N} p_i = 1$. Define a new popularity distribution $\mathcal{P}_2 = \{1 - \frac{N_1}{KM_r}, \frac{1}{KM_r}, \frac{1}{KM_r}, ..., \frac{1}{KM_r}\}$ over the files $\mathcal{F}_1$. In other words, compared to $(K,\mathcal{F}_1,\mathcal{P}_1)$, in this new system, each non-empty file is requested with a smaller probability $\frac{1}{KM_r}$. Intuitively, its expected transmission rate should be even lower, which is stated below.

Lemma 2: Let $R(K,\mathcal{F}_1,\mathcal{P}_2)$ be the minimum expected rate required to meet the requests by K users, each of which randomly requests a file in $\mathcal{F}_1$ according to the popularity distribution $\mathcal{P}_2$. We have

$$R(K,\mathcal{F}_1,\mathcal{P}_1) \geq R(K,\mathcal{F}_1,\mathcal{P}_2). \tag{11}$$

This lemma is similar to the uniformization argument in the proof of Claim 2 in [9]. The proof can also use a similar coupling idea as in Lemma 1. For completeness, we provide the proof in the appendix.

With Lemma 1 and Lemma 2, we have proved that $R(K,\mathcal{F},\mathcal{P}) \geq R(K,\mathcal{F}_1,\mathcal{P}_2)$. In the following analysis for the first part of Theorem 1, we will focus on $R(K,\mathcal{F}_1,\mathcal{P}_2)$. Note that the system $(K,\mathcal{F}_1,\mathcal{P}_2)$ is precisely the "System 2" that we discussed in Section III-A. Next, we will derive a lower bound on the average transmission rate of System 2, which also provides a lower bound on $R(K,\mathcal{F},\mathcal{P})$. We will derive this lower bound on the *average* transmission rate of System 2 by relating it to a lower bound on the *worst-case* transmission rate. Note that since all files have equal popularity in System 2, the fact that its *average* transmission rate is at most a constant factor away from its *worst-case* transmission rate is in fact known from the results in [9] and [13]. For example, we can obtain a lower bound on the average transmission rate of System 2 from Theorem 2 in [9] by choosing $c = 1$, $N_l = N_1$ there and by choosing K_l in [9] as the number of users requesting popular files. However, the lower bound derived in this way involves an expectation over K_l. Since later we will use System 2 again to deal with unpopular files, we wish to obtain a lower bound that is a function of the total number of users K. The following derivation accounts for such technical details, and at the same time yields the factor $1/11$ in (6). (In contrast, an explicit expression for such a factor is not provided in [13]). We note that some of the reduction techniques below and in Sections IV-B to IV-C are also similar to [9][13], although here we exploit the fact that $p_{N_1} \approx \frac{1}{KM_r}$ to obtain the $1/11$ factor (see Proposition 1 below).

Towards this end, rather than only proving the lower bound for System 2, we instead prove a more general result that will be used again in Subsections C-E. Specifically, consider a system $(K,\mathcal{F}',\mathcal{P}')$. Its file set is $\mathcal{F}' = \{F_0, F_1, ..., F_{N_0}\}$, and the corresponding popularity distribution is $\mathcal{P}' = \{p_0, p, ..., p\}$, where $p \leq \frac{1}{KM_r}$ and $p_0 = 1 - N_0 p \geq 0$. Note that $(K,\mathcal{F}',\mathcal{P}')$ becomes the "System 2" when $N_0 = N_1$ and $p = \frac{1}{KM_r}$.

Proposition 1: With K users requesting files independently in $\mathcal{F}'$ according to the corresponding popularity distribution $\mathcal{P}'$, the lower bound on the expected transmission is

$$R(K,\mathcal{F}',\mathcal{P}') \geq \frac{1}{11} Kp \cdot (N_0 - M), \tag{12}$$

for any $M \geq 0$.

From now till Subsection C, we will prove Proposition 1.

Note that in the system $(K,\mathcal{F}',\mathcal{P}')$, it is possible that some file is requested by multiple users. In Section IV-B, we will reduce it to the third system where every non-empty requested file is requested exactly once. Towards that end, we first characterize the number of distinct files requested in system $(K,\mathcal{F}',\mathcal{P}')$.

For a given system setting $(K,\mathcal{F}',\mathcal{P}')$, let $I_i = 1$ if user i requests a non-empty file, and let $I_i = 0$ if user i requests the empty file. Denote $K_r = \sum_{i=1}^{K} I_i$. Then, K_r is the number of users who request non-empty files. All I_i are *i.i.d.* distributed with mean $N_0 p$. The probability distribution for K_r is given by

$$P(K_r = K_1) = C_K^{K_1} \left(N_0 p\right)^{K_1} \left(1 - N_0 p\right)^{K - K_1}.$$

Lemma 3: Define $K_1 \triangleq \lfloor K N_0 p \rfloor$. Since $p \leq \frac{1}{KM_r}$, we have $K_1 \leq \lfloor \frac{N_0}{M_r} \rfloor$, and

$$P(K_r \geq K_1) \geq \frac{1}{2}. \tag{13}$$

This follows from the result in [17], which shows that any median must lie in the interval $[\lfloor np \rfloor, \lceil np \rceil]$, for a binomial distribution $B(n,p)$.

In other words, with probability no less than 0.5, no less than K_1 users request non-empty files. Still, some of these K_1 users may request a common file. Next, we are interested in the number of distinct files that are requested. Denote this number as K_d.

Lemma 4: Given that there are K_r users requesting non-empty files, the probability that K_d (i.e., the number of distinct files requested) is no smaller than $\min\{\lfloor \frac{1}{2} K_r \rfloor, \lfloor \frac{1}{2} K_1 \rfloor\}$ is greater than or equal to 0.91.

Proof: Clearly, we only need to consider $K_r \leq K_1$ (because a larger value of K_r only increases the number of distinct files). When $K_r = 1, 2$, or 3, we have that $\lfloor \frac{1}{2} K_r \rfloor$ equals 0 or 1. In this case, it is easy to see that this lemma holds, since there must be at least one distinct file requested.

For $K_r \geq 4$, consider only those K_r users requesting non-empty files. Each user requests one file from the N_0 non-empty files uniformly randomly and independently. There are $N_0^{K_r}$ possible request patterns for the K_r users, each of which is equally likely. For some of these request patterns, the number of distinct files are smaller than $K_2 \triangleq \lfloor \frac{1}{2} K_r \rfloor$. The number of such request patterns must be smaller than $C_{N_0}^{K_2} \cdot K_2^{K_r}$. To see this, note that the first term is the number of ways to choose K_2 files from the N_0 non-empty files. The second term is the number of ways that each user can choose one of the K_2 files. We thus have

$$\begin{aligned} P(K_d \leq K_2 | K_r) <& \frac{C_{N_0}^{K_2} K_2^{K_r}}{N_0^{K_r}} \\ \leq& \frac{e\sqrt{N_0}(\frac{N_0}{e})^{N_0}}{\sqrt{2\pi(N_0 - K_2)}(\frac{N_0 - K_2}{e})^{N_0 - K_2}} \\ & \cdot \frac{1}{\sqrt{2\pi K_2}(\frac{K_2}{e})^{K_2}} \left(\frac{K_2}{N_0}\right)^{K_r} \\ \leq& \frac{e}{2\pi} \left(\frac{N_0}{N_0 - K_2}\right)^{N_0 - K_2} \cdot (\frac{N_0}{K_2})^{K_2 - K_r}. \end{aligned}$$

Here, we have used Stirling's formula in the second step, i.e.,

$$\sqrt{2\pi n} \left(\frac{n}{e}\right)^n \leq n! \leq e\sqrt{n} \left(\frac{n}{e}\right)^n.$$

In the third step, we have used $\sqrt{\frac{N_0}{K_2(N_0 - K_2)}} \leq 1$, due to $K_2 \geq 2$ and $K_2 \leq \frac{1}{2} N_0$. It is easy to prove that $(1+x)^{\frac{1}{x}} \leq e$ for any $x > 0$. Therefore,

$$\left(\frac{N_0}{N_0 - K_2}\right)^{N_0 - K_2} = \left(1 + \frac{K_2}{N_0 - K_2}\right)^{\frac{N_0 - K_2}{K_2} \cdot K_2} \leq e^{K_2}.$$

Recall that at the beginning of the proof, we have restricted our attention to $K_r \leq K_1$ and $K_r \geq 4$. Since $K_2 \triangleq \lfloor \frac{1}{2} K_r \rfloor$, we have $2 \leq K_2 \leq \frac{1}{2} K_r$. Hence, using Lemma 3, we have

$K_r \leq K_1 \leq \lfloor \frac{N_0}{M_r} \rfloor \leq \frac{1}{3}N_0$. Therefore, we have

$$\begin{aligned} P(K_d \leq K_2|K_r) &< \frac{e}{2\pi} e^{K_2} \cdot \left(\frac{N_0}{K_2}\right)^{K_2 - K_r} \\ &\leq \frac{e}{2\pi} e^{K_2} \cdot 6^{K_2 - K_r} \\ &< \frac{e}{2\pi} e^{2K_2 - K_r} \cdot \left(\frac{6}{e}\right)^{K_2 - K_r} \\ &\leq \frac{e}{2\pi} \left(\frac{6}{e}\right)^{-2}, \end{aligned}$$

where the third inequality is due to $K_2 \leq \frac{1}{2}K_r \leq \frac{1}{6}N_0$, and the fourth inequality is due to $K_2 \leq \frac{1}{2}K_r$. Finally, $P(K_d > K_2|K_r) = 1 - P(K_d \leq K_2|K_r) > 0.91$. ■

B. Reduction Step 3

Combing Lemma 3 and Lemma 4, we can show that, with probability no less than 0.455, the number of distinct files requested is no smaller than $\lfloor \frac{1}{2}K_1 \rfloor$. We now perform the third reduction. For $\mathbb{W}(K, \mathcal{F}', \mathcal{P}')$, we divide all the possible request patterns into two subsets. The first subset contains those request patterns such that the number of users requesting distinct files is no smaller than $K_3 \triangleq \lfloor \frac{K_1}{2} \rfloor$. The other request patterns constitute the second subset. The sum probability for the second subset is denoted as $\pi(K_1)$.

We then construct the third system as follows: in the third system, with probability $\pi(K_1)$ the users will all request the empty file. With probability $1 - \pi(K_1)$, exactly K_3 users will request exactly K_3 distinct non-empty files from $F_1, ..., F_{N_0}$, and all other users will request the empty file. Note that there are exactly $C_K^{K_3} A_{N_0}^{K_3}$ request patterns where exactly K_3 users request K_3 distinct non-empty files. We let each such request pattern occur with equal probability $\frac{1-\pi(K_1)}{C_K^{K_3} A_{N_0}^{K_3}}$.

Let this third system be denoted by $\mathbb{W}_3(K_3, K_1)$, and let $R(K, \mathbb{W}_3)$ be the corresponding minimum expected transmission rate. Then. we have the following lemma.

Lemma 5:

$$R(K, \mathcal{F}', \mathcal{P}') \geq R(K, \mathbb{W}_3). \tag{14}$$

Proof: The proof uses coupling [19]. For every $W_i \in \mathbb{W}(K, \mathcal{F}', \mathcal{P}')$, map it to a random $W_i^{'} \in \mathbb{W}_3(K_3, K_1)$ as follows. If the number of users requesting non-empty files in W_i is less than K_1, or the number of distinct non-empty files requested is less than $K_3 \triangleq \lfloor \frac{K_1}{2} \rfloor$, then in $W_i^{'}$ all users request the empty file. Otherwise, we perform the mapping described below.

For every remaining W_i with $K_d \geq K_3$, we conduct the following splitting procedure.

- For each non-empty file that is requested by some users, randomly choose one user requesting it. Note that there are K_d such chosen users.
- Among the chosen users, randomly choose K_3 of them. These K_3 users now request distinct non-empty files, and we let all other users request the empty file.

Given any cache placement and transmission scheme $\mathfrak{F}$, since $W_i^{'}$ requests a subset of the files in W_i, we have

$$r_{\mathfrak{F}}(K, W_i) \geq r_{\mathfrak{F}}(K, W_i^{'}). \tag{15}$$

It remains to show that, if W_i is chosen according to the distribution of $\mathbb{W}(K, \mathcal{F}_1, \mathcal{P}_2)$, then the resulting $W_i^{'}$ has the same distribution as that of request patterns in $\mathbb{W}_3(K_3, K_1)$. To see this, note that the probability with which W_i' requests no empty files is exactly $1 - \pi(K_1)$. Further, due to symmetry on the files and the users in $\mathbb{W}(K, \mathcal{F}_1, \mathcal{P}_2)$, along with the symmetry of our mapping, each pattern $W_i^{'}$ that requests non-empty files must occur with equal probability, i.e., there are $C_K^{K_3} A_{N_0}^{K_3}$ request patterns each of equal probability, that sums to the probability $1 - \pi(K_1)$. We then conclude that each W_i' occurs with the same probability as in $\mathbb{W}_3(K_3, K_1)$.

Thus, with the coupling method [19], we have

$$R_{\mathfrak{F}}(K, \mathcal{F}', \mathcal{P}') \geq R_{\mathfrak{F}}(K, \mathbb{W}_3). \tag{16}$$

and the result then follows. ■

C. Reduction Step 4 & the Lower Bound for Popular files

We now consider the 4th system $\mathbb{W}_4$. In this system, there are always $K_3 \triangleq \lfloor \frac{K_1}{2} \rfloor$ users requesting K_3 distinct non-empty files and all the other users request the empty file. We have

Lemma 6: $R(K, \mathbb{W}_3) = (1 - \pi(K_1))R(K, \mathbb{W}_4)$.

Proof: Recall that in the third system $\mathbb{W}_3(K_3, K_1)$, with probability $\pi(K_1)$ the users all request the empty file, which requires zero rate. With probability $1 - \pi(K_1)$, exactly K_3 users request exactly K_3 distinct files. Each such pattern occurs with probability $\frac{1-\pi(K_1)}{C_K^{K_3} A_{N_0}^{K_3}}$. In the fourth system $\mathbb{W}_4$, all request patterns have exactly K_3 users requesting exactly K_3 distinct files, and each pattern occurs with probability $\frac{1}{C_K^{K_3} A_{N_0}^{K_3}}$. Therefore, the rate needed for the third system is a linear combination of zero rate (with portion $\pi(K_1)$) and the rate needed for the fourth system (with portion $1 - \pi(K_1)$). The lemma then follows. ■

Next we focus on the system $\mathbb{W}_4$.

Let H_i, $i = 1, 2, ..., C_K^{K_3}$ be the $C_K^{K_3}$ choices of picking K_3 users out of the K users. In system $\mathbb{W}_4$, if in a request W_j, the users requesting distinct non-empty files are exactly in H_i, we denote it by $W_j \in \overline{H_i}$. In other words, $\overline{H_i}$ is the set of request patterns in which the users requesting non-empty files is exactly the same as in H_i. Note that there are $A_{N_0}^{K_3} = \frac{N_0!}{(K-K_3)!}$ such patterns in each $\overline{H_i}$. We have the following result.

Lemma 7: Consider systems $\mathbb{W}_4$ where there are always exactly K_3 users requesting distinct files in $\mathcal{F}'$ and the other $K - K_3$ users request the empty file. For any H_i, $i = 1, 2, ..., C_K^{K_3}$, the following holds,

$$\sum_{W_j \in \overline{H_i}} r_{\mathfrak{F}}(K, W_j) \geq A_{N_0}^{K_3} \cdot \frac{\left\lfloor \frac{N_0}{K_3} \right\rfloor K_3 - K_3 M}{\left\lfloor \frac{N_0}{K_3} \right\rfloor}. \tag{17}$$

Note that Lemma 7 immediately implies that

$$R(K, \mathbb{W}_4) \geq \frac{\left\lfloor \frac{N_0}{K_3} \right\rfloor K_3 - K_3 M}{\left\lfloor \frac{N_0}{K_3} \right\rfloor}. \tag{18}$$

Proof: Without loss of generality, suppose that $H_i = \{1, 1, ..., 1, 0, 0, ..., 0\}$. In other words, user 1, 2,..., K_3 are requesting distinct non-empty files. Each user has a cache, labeled $M_1, M_2, ..., M_{K_3}$, each of which has a common storage size M.

There are $N_0!$ permutations for the N_0 files. For each permutation, we split it into $\lfloor \frac{N_0}{K_3} \rfloor$ subgroups, each with K_3 files. Denote $r(i,j)$ as the rate needed to meet the users' requests if their request pattern is the same as the j-th subgroup in the i-th permutation, i.e., when the k-th user requests the k-th file in the subgroup, $k = 1, 2, ..., K_3$.

For each permutation i, consider all the sub-groups (i.e., request patterns) as a whole. Recall that the cache content is fixed when these request patterns vary. Consider a feasible cache placement and transmission scheme $\mathfrak{F}$. Based on the cached content $M_1, ..., M_{K_3}$, and the transmissions from the server for each request pattern (with rates $r(i,1), ..., r(i, \lfloor \frac{N_0}{K_3} \rfloor)$, respectively), the K_3 users together must be able to reconstruct all $K_3 \cdot \lfloor \frac{N_0}{K_3} \rfloor$ files. Hence,

$$\sum_{j=1}^{\lfloor \frac{N_0}{K_3} \rfloor} r_{\mathfrak{F}}(i,j) + \sum_{k=1}^{K_3} M_k \geq K_3 \cdot \left\lfloor \frac{N_0}{K_3} \right\rfloor. \tag{19}$$

Summarizing over all $N_0!$ permutations, we have

$$\sum_{i=1}^{N_0!} \sum_{j=1}^{\lfloor \frac{N_0}{K_3} \rfloor} r_{\mathfrak{F}}(i,j) \geq \left(\left\lfloor \frac{N_0}{K_3} \right\rfloor \cdot K_3 - K_3 M \right) \cdot N_0!. \tag{20}$$

Note that there are $A_{N_0}^{K_3}$ request patterns $W_j \in \overline{H_i}$, while there are $\left\lfloor \frac{N_0}{K_3} \right\rfloor \cdot N_0!$ subgroups among all the $N_0!$ permutations. By symmetry[2], each $W_j \in \overline{H_i}$ appears an equal number of times in these subgroups. Hence, the number of times each $W_j \in \overline{H_i}$ appears in the summation in Equation (20) is $\frac{\left\lfloor \frac{N_0}{K_3} \right\rfloor \cdot N_0!}{A_{N_0}^{K_3}}$. Hence,

$$\begin{aligned} \frac{\sum_{W_j \in \overline{H_i}} r_{\mathfrak{F}}(K, W_j)}{A_{N_0}^{K_3}} &= \frac{1}{\left\lfloor \frac{N_0}{K_3} \right\rfloor \cdot N_0!} \cdot \sum_{i=1}^{N_0!} \sum_{j=1}^{\lfloor \frac{N_0}{K_3} \rfloor} r_{\mathfrak{F}}(i,j) \\ &\geq \frac{1}{\left\lfloor \frac{N_0}{K_3} \right\rfloor \cdot N_0!} \cdot N_0!(\left\lfloor \frac{N_0}{K_3} \right\rfloor K_3 - K_3 M) \\ &= \frac{\left\lfloor \frac{N_0}{K_3} \right\rfloor K_3 - K_3 M}{\left\lfloor \frac{N_0}{K_3} \right\rfloor}. \end{aligned} \tag{21}$$

We therefore conclude this lemma. ■

Denote the right hand side of Equation (18) by $f(K_3)$. From Lemmas 5 and 6, the minimum expected rate can be bounded by

$$\begin{aligned} R(K, \mathcal{F}', \mathcal{P}') &\geq R(K, \mathbb{W}_3) \\ &= (1 - \pi(K_1)) \cdot R(K, \mathbb{W}_4) \\ &\geq (1 - \pi(K_1)) f(K_3). \end{aligned} \tag{22}$$

Recall that $K_1 \triangleq \lfloor K N_0 p \rfloor \leq \lfloor \frac{N_0}{M_r} \rfloor$ and $K_3 \triangleq \lfloor \frac{1}{2} K_1 \rfloor$. We now consider two cases.

If $K N_0 p \leq 5$, it is easy to verify that

$$\begin{aligned} f(K_3) &\geq f(1) \\ &= 1 - \frac{M}{N_0} \\ &= \frac{1}{N_0}(N_0 - M) \\ &\geq \frac{1}{5} K p (N_0 - M). \end{aligned} \tag{23}$$

Next, we consider the case $K N_0 p > 5$. By definition, we have

$$\begin{aligned} K_3 &= \left\lfloor \frac{1}{2} K_1 \right\rfloor \\ &\geq \frac{1}{2} K_1 - \frac{1}{2} \quad \text{(since } K_1 \text{ is an integer)} \\ &= \frac{1}{2} \lfloor K N_0 p \rfloor - \frac{1}{2} \\ &\geq \frac{1}{2}(K N_0 p - 1) - \frac{1}{2} \\ &= \frac{1}{2} K N_0 p - 1 \geq 0.3 K N_0 p, \end{aligned} \tag{24}$$

where in the last step we have used $K N_0 p > 5$. Since $M_r \geq 3$, we have

$$\begin{aligned} f(K_3) &= K_3 - \frac{K_3 M}{\left\lfloor \frac{N_0}{K_3} \right\rfloor} \\ &\geq K_3 \left(1 - \frac{M}{\frac{N_0}{K_3} - 1} \right) \geq K_3 \left(1 - \frac{M}{\frac{10}{3Kp} - 1} \right) \\ &\geq K_3 \left(1 - \frac{M}{\frac{3}{Kp}} \right) \quad (\text{using} \frac{1}{Kp} \geq M_r \geq 3) \\ &\geq (0.5 K N_0 p - 1) \cdot \left(1 - \frac{1}{3} M K p \right) \\ &\geq \frac{1}{5} K p \left(N_0 - M \right), \end{aligned} \tag{25}$$

where the last inequality is due to

$$\begin{aligned} (0.5 K N_0 p - 1) \cdot \left(1 - \frac{1}{3} M K p \right) - \frac{1}{5} K p \left(N_0 - M \right) & \\ \geq (0.5 K N_0 p - 1) \frac{2}{3} - \frac{1}{5} K p (N_0 - M) & \\ \geq \frac{2}{15} K N_0 p - \frac{2}{3} + \frac{1}{5} K p M & \\ > 0, & \end{aligned} \tag{26}$$

where the first inequality is due to $KMp \leq 1$, and the third inequality is due to $K N_0 p > 5$ and $KMp \geq 0$.

Using both (23) and (25) into (22), we conclude that the minimum expected rate needed for $\mathbb{W}(K, \mathcal{F}, \mathcal{P})$ is bounded by

$$\begin{aligned} R(K, \mathcal{F}', \mathcal{P}') &\geq 0.455 \cdot \frac{1}{5} K p \left[N_0 - M \right]_+ \\ &\geq \frac{1}{11} K p \left[N_0 - M \right]_+. \end{aligned} \tag{27}$$

Therefore, we have proved Proposition 1.

Applying Proposition 1 to System 2 $(K, \mathcal{F}_1, \mathcal{P}_2)$ (i.e., with $N_0 = N_1$ and $p = \frac{1}{K M_r}$), we immediately have the following lower bound for the expected transmission rate of System 2:

$$R(K, \mathcal{F}_1, \mathcal{P}_2) \geq \frac{1}{11 M_r}(N_1 - M). \tag{28}$$

[2] We note that this is similar to the symmetrization argument in [9].

D. Lower Bound Accounting for Both Popular & Unpopular Files

The earlier proof only considers the more "popular" files, i.e., files with popularity larger than a threshold. Now we will move our attention to the unpopular files and prove the first term in the lower bound 6, i.e., $R(K,\mathcal{F},\mathcal{P}) \geq \frac{1}{11M_r}[N_1+N_2-M]_+$. To the best of our knowledge, this treatment of unpopular files has not been reported in the literature. Intuitively, depending on the system setting, the lower bound may be dominated by either the popular files or the unpopular files. Thus, we believe that our capability to quantify the impact of unpopular files is the key reason that we can obtain the improved constant-factor characterization in this paper, even in non-asymptotic settings.

Interestingly, readers will see soon that we will re-apply the results for System 2 constructed in Section IV.A. Recall that in System 2 all users either request one of the non-empty files with a common popularity that is no less than $1/KM_r$, or request the empty file. For the unpopular files, their popularity is lower than $1/KM_r$, and hence we cannot directly use the results for System 2. However, next we introduce a new "merging" idea that will eventually allow us to re-apply the results for System 2. Specifically, we will merge several unpopular files into one file, so that the popularity of the new file is no less than $1/KM_r$. We will show that this merging step will only lower the achievable rate for serving unpopular files. Thus, in the end we obtain a new system similar to System 2, from which a lower bound for the achievable rate of the original system can be derived. The detail analysis is as follows.

Recall that the original set of files is $\mathcal{F}$, and the corresponding popularity distribution is $\mathcal{P}$. Consider another system, where the set of files is $\mathcal{F}_3 = \{F_0, F_1, ..., F_{N_1}, F_{N_1+1}, F_{N_1+2}, ..., F_N\}$ (recall that F_0 is again the empty file). The corresponding popularity distribution is $\mathcal{P}_3 = \{p_0^{'}, \frac{1}{KM_r}, ..., \frac{1}{KM_r}, p_{N_1+1}, p_{N_1+2}, ..., p_N\}$ where $p_0^{'} = 1 - \frac{N_1}{KM_r} - \sum_{i=N_1+1}^{N} p_i$. In other words, we decrease the popularity of files $F_1, F_2, ..., F_{N_1}$ in the original system to $\frac{1}{KM_r}$. Following exactly the same way as the proof of Lemma 2, we can prove that

$$R(K,\mathcal{F},\mathcal{P}) \geq R(K,\mathcal{F}_3,\mathcal{P}_3). \tag{29}$$

Again, we will perform a series of further reductions and finally construct a system with a smaller rate, which can utilize the results in previous analysis of "System 2".

To proceed, we need the lemma below. Denote a file set by $\mathcal{T}_1 = \{T_1, T_2, ..., T_t\}$. Each element T_i can either be a regular file or the empty file. Let its corresponding popularity distribution be $\mathcal{Q}_1 = \{q_1, q_2, ..., q_t\}$, where $\sum_{i=1}^{t} q_i = 1$. Denote another file set by $\mathcal{T}_2 = \{T_1, T_2, ..., T_{t-2}, T_{t+1}\}$ with popularity distribution $\mathcal{Q}_2 = \{q_1, q_2, ..., q_{t-2}, q_{t+1}\}$. Here $q_{t+1} = q_{t-1} + q_t$. In other words, the two files T_{t-1} and T_t in the first system are replaced by *one* file T_{t+1} in the second system. Intuitively, it should be easier (i.e., requiring less cache and lower transmission rate) to serve the second system because there is less "diversity". This statement is made precise below.

Lemma 8: Let $R(K,\mathcal{T}_1,\mathcal{Q}_1)$ be the minimum expected rate required to meet the requests by K users, each of which randomly requests a file in $\mathcal{T}_1$ according to the popularity distribution $\mathcal{Q}_1$. Let $R(K,\mathcal{T}_2,\mathcal{Q}_2)$ be defined similarly for $\mathcal{Q}_2$. We have

$$R(K,\mathcal{T}_1,\mathcal{Q}_1) \geq R(K,\mathcal{T}_2,\mathcal{Q}_2). \tag{30}$$

Proof: The request set for $(K,\mathcal{T}_1,\mathcal{Q}_1)$ is $\mathbb{W}(K,\mathcal{T}_1,\mathcal{Q}_1) = \{W_i\}$, where $W_i = \{f_{i1}, f_{i2}, ..., f_{iK}\}$ and $f_{ij} \in \mathcal{T}_1$. We first construct a mapping from $\mathbb{W}(K,\mathcal{T}_1,\mathcal{Q}_1)$ to $\mathbb{W}(K,\mathcal{T}_2,\mathcal{Q}_2)$.

For every request $W_i \in \mathbb{W}(K,\mathcal{T}_1,\mathcal{Q}_1)$, we map it to a request $W_i^{'} \in \mathbb{W}(K,\mathcal{T}_2,\mathcal{Q}_2)$ as follows. If file T_{t-1} or T_t in $\mathcal{T}_1$ is requested in W_i, we replace it by T_{t+1}.

For a cache placement and transmission scheme $\mathfrak{F}$, suppose that each user can retrieve the file requested in W_i with rate $r_{\mathfrak{F}}(K,W_i)$. Using the same $\mathfrak{F}$, with the replacement of T_{t+1} for T_{t-1} or T_t in both cache placement and transmissions, the rate $r_{\mathfrak{F}}(K,W_i)$ must be able to satisfy the request of $W_i^{'}$. Therefore, we have

$$r_{\mathfrak{F}}(K,W_i) \geq r_{\mathfrak{F}}(K,W_i^{'}). \tag{31}$$

Further, if W_i follows the distribution with which patterns are requested in $\mathbb{W}(K,\mathcal{T}_1,\mathcal{Q}_1)$, $W_i^{'}$ will follow the distribution with which patterns are requested in $\mathbb{W}(K,\mathcal{T}_2,\mathcal{Q}_2)$. To see this, consider any request pattern $\{f_1, ..., f_K\}$. Fix a user k. If $f_k \neq T_{t+1}$, the probability that user k requests f_k in the mapped request pattern $W_i^{\prime}$ is exactly the probability that user k requests f_k in $\mathbb{W}(K,\mathcal{T}_2,\mathcal{Q}_2)$. If $f_k = T_{t+1}$, note that user k requesting T_{t+1} in $W_i^{\prime}$ corresponds to either T_t or T_{t-1} being requested in W_i. Hence, the probability that user k in the mapped request pattern $W_i^{\prime}$ requests file T_{t+1} is equal to $p_t + p_{t-1} = p_{t+1}$, which is also the same as the probability that user k requests T_{t+1} in $\mathbb{W}(K,\mathcal{T}_2,\mathcal{Q}_2)$. Finally, note that the probability with which a pattern is requested is the product of the probability that each file in the pattern is requested by the corresponding user. Therefore, $W_i^{\prime}$ must have the same distribution as in system $\mathbb{W}(K,\mathcal{T}_2,\mathcal{Q}_2)$. Thus, by the coupling method [19], we must have

$$R_{\mathfrak{F}}(K,\mathcal{T}_1,\mathcal{Q}_1) \geq R_{\mathfrak{F}}(K,\mathcal{T}_2,\mathcal{Q}_2). \tag{32}$$

The results of this lemma then follows. ■

Next, we create a new system $(K,\mathcal{F}_4,\mathcal{P}_4)$ originated from $(K,\mathcal{F}_3,\mathcal{P}_3)$, by merging multiple files in $\mathcal{F}_3$ to a new file in $\mathcal{F}_4$ (described below) and combining their popularity (similar to the mapping from $\mathcal{Q}_1$ to $\mathcal{Q}_2$). We denote this new file set as $\mathcal{F}_4 = \{F_0, F_1, ..., F_{N_1}, V_1, ..., V_{N_2}\}$ and the popularity distribution as $\mathcal{P}_4 = \{v_0, \frac{1}{KM_r}, ..., \frac{1}{KM_r}, v_1, v_2, ...v_{N_2}\}$. Here, F_0 is again the empty file and $v_0 = 1 - \frac{N_1}{M_r} - \sum_{i=1}^{N_2} v_i$. The other files $V_1, ..., V_{N_2}$ are non-empty files and we pick them in such a way that they all have similar popularity $v_i \approx 1/KM_r$. Specifically, recall that $\frac{1}{KM_r} > p_{N_1+1} \geq p_{N_1+2} \geq ... \geq p_N$. Let $h_0 = 0$. Pick h_1 as the smallest integer such that $\sum_{j=N_1+1}^{N_1+h_1} p_j \geq 1/KM_r$. We then replace files $F_{N_1+1}, ..., F_{N_1+h_1}$ by one file V_1, whose popularity is $v_1 = \sum_{j=N_1+1}^{N_1+h_1} p_j$. Similarly, for each $i = 2, 3, ...,$ we pick h_i

as the smallest integer such that

$$\sum_{j=N_1+h_{i-1}+1}^{N_1+h_i} p_j \geq \frac{1}{KM_r} \tag{33}$$

and then replace files $F_{N_1+h_{i-1}+1}, ..., F_{N_1+h_i}$ by one file V_i, whose popularity is $v_i = \sum_{j=N_1+h_{i-1}+1}^{N_1+h_i} p_j$. In this way, each non-empty file's popularity satisfies $1/KM_r \leq v_i \leq 2/KM_r$ for all $1 \leq i \leq N_2$. Note that after the merging steps finish, there could be some files unmerged whose sum popularity is smaller than $\frac{1}{KM_r}$. Hence, we have $\sum_{i>N_1} Kp_i < N_2 \cdot \frac{2}{KM_r} + \frac{1}{KM_r}$, which can turn into

$$N_2 > \frac{\sum_{i>N_1} p_i}{2/KM_r} - \frac{1}{2}. \tag{34}$$

Since N_2 is an integer, we must have

$$\begin{aligned} N_2 &\geq \left\lfloor \frac{\sum_{i>N_1} p_i}{2/KM_r} - \frac{1}{2} \right\rfloor + 1 \\ &= \left\lfloor \frac{\sum_{i>N_1} p_i}{2/KM_r} + \frac{1}{2} \right\rfloor. \end{aligned} \tag{35}$$

In the following, we will only take $\left\lfloor \frac{\sum_{i>N_1} p_i}{2/KM_r} + \frac{1}{2} \right\rfloor$ of these merged files, i.e., let $N_2 = \left\lfloor \frac{\sum_{i>N_1} p_i}{2/KM_r} + \frac{1}{2} \right\rfloor$. Let this new system be $(K, \mathcal{F}_4, \mathcal{P}_4)$.

By applying Lemma 8 iteratively, we can show that

$$R(K, \mathcal{F}_3, \mathcal{P}_3) \geq R(K, \mathcal{F}_4, \mathcal{P}_4). \tag{36}$$

Define $\mathcal{P}_5 = \{1 - \frac{N_1+N_2}{KM_r}, \frac{1}{KM_r}, ..., \frac{1}{KM_r}\}$ over file set $\mathcal{F}_4$. In other words, the first file (which corresponds to the empty file) is requested with probability $1 - \frac{N_1+N_2}{KM_r}$, and the other $(N_1 + N_2)$ files are requested with probability $\frac{1}{KM_r}$ each. Similar to Lemma 2, we can prove that

$$R(K, \mathcal{F}_4, \mathcal{P}_4) \geq R(K, \mathcal{F}_4, \mathcal{P}_5). \tag{37}$$

Now, note that the system $(K, \mathcal{F}_4, \mathcal{P}_5)$ is of the same form as the system $(K, \mathcal{F}_1, \mathcal{P}_2)$: all non-empty files are requested with a common probability that is greater than or equal to $\frac{1}{KM_r}$ (this is also of the form of the "System 2" that we referred to in Sections III-A and IV-A). With System $(K, \mathcal{F}_4, \mathcal{P}_5)$, we can directly apply Proposition 1 and obtain

$$R(K, \mathcal{F}_4, \mathcal{P}_5) \geq \frac{1}{11M_r} [N_1 + N_2 - M]_+. \tag{38}$$

Remark: We believe that the above characterization for the unpopular files is crucial for obtaining the improved constant-factor results that hold for arbitrary distributions and system settings. Again take Zipf distribution with $\alpha = 1$ as an example. Suppose that the number of files N is large. As we discussed at the beginning of this subsection, it is easy to see that $p_i \approx \frac{1}{i \log N}$, and thus the threshold is $N_1 \approx \frac{KM}{\log N}$. On the other hand, the number of virtual files merged from the unpopular files is about $N_2 = \Theta(\frac{\frac{\log N - \log N_1}{\log N}}{2/KM})$. From our earlier results, the lower bound due to popular files is $\frac{1}{11}(\frac{N_1}{M} - 1) \approx \frac{1}{11}(\frac{K}{\log N} - 1)$, while the lower bound due to unpopular files is about $\frac{1}{11}\frac{N_2}{M} \approx \frac{1}{22}\frac{K}{\log N} \cdot \log \frac{N \log N}{KM}$. Note that depending on the relationship between N and K, the term due to unpopular files may be larger or smaller than the term due to popular files. For instance, if we keep N fixed and let $K \to \infty$, then the term due to unpopular files will be dominated by the term due to popular files. Such a setting has been studied in Corollary 1 of [13]. However, in general $N/(KM)$ could be large, and thus the unpopular files may dominate. In that case, if we did not use the characterization in this subsection, we would be unable to obtain the sharper results in this paper.

E. Further Refining the Lower Bound

In (38), we have established the first term of (6). However, the first term of (6) may becomes zero when $N_1 + N_2 \leq M$. In that case, the first term of (6) will be too loose to be a lower bound for the average transmission rate. What happens is that the cache space M is abundant, and hence more than $N_1 + N_2$ files can be accommodated in the cache. Intuitively, we should "relax" the threshold N_1 for "popular files", so that a larger number of files can be considered for caching. Towards this end, we now take p_{N_x} as the popularity threshold. There are N_x files with popularity no smaller than p_{N_x}.

Lemma 9: With K users requesting files independently in $\mathcal{F}$ according to the corresponding popularity distribution $\mathcal{P}$, the lower bound on the expected transmission rate is given by

$$R(K, \mathcal{F}, \mathcal{P}) \geq \frac{1}{11} K p_{N_x} (N_x - M), \tag{39}$$

for any N_x satisfying $p_{N_x} \leq \frac{1}{KM_r}$.

This lemma can be obtained by reducing the popularity of files F_i $(1 \leq i \leq N_x)$ in the original file-set $\mathcal{F}$ to p_{N_x} and applying Proposition 1. We note that the use of N_x here is similar to the use of l in Theorem 2 of [13], which allows a larger number of files to be considered for caching. However, the use of N_y introduced below again accounts for files less popular than p_{N_x}, which is not reported in [13].

Then, if we apply the merging technique introduced in subsection D, we can construct N_y files with popularity p_{N_x} from files F_{N_x+1},F_{N_x+2},...,F_N, where $N_y = \left\lfloor \frac{\sum_{i>N_x} p_i}{2/p_{N_x}} + \frac{1}{2} \right\rfloor$. Applying Proposition 1, we have the following lemma.

Lemma 10: With K users requesting files independently in $\mathcal{F}$ according to the corresponding popularity distribution $\mathcal{P}$, the lower bound on the expected transmission rate is given by

$$R(K, \mathcal{F}, \mathcal{P}) \geq \frac{1}{11} K p_{N_x} (N_x + N_y(N_x) - M), \tag{40}$$

for any N_x satisfying $p_{N_x} \leq \frac{1}{KM_r}$.

Combining Equations (38) and (40), the result of Theorem 1 then follows.

V. Upper Bound on Expected Transmission Rate

In this section, we show that, by combing RLFU with another scheme that divides the files into 3 groups, we can obtain a tight achievable bound. Recall the following statement of Theorem 2,

$$R(K,\mathcal{F},\mathcal{P}) \le \min\Big(\frac{[N_1-M]_+}{\max(1,M)} + \sum_{i>N_1} Kp_i, \qquad Kp_{N_3}(N_3-M) + \sum_{i>N_3} Kp_i\Big), \tag{41}$$

where $N_3 = \lfloor M \rfloor + 1$.

Remark: We note that the first term is similar to the upper bound in [13]. However, the denominator is $\max(1,M)$ instead of M. Further, the second term of (41) is new. These changes are important to deal with the case when $N_1 < M$ and when M is small. Please see scheme 2 and scheme 3 in the proof below for details.

Proof of Theorem 2: We prove by considering 3 schemes. We first consider scheme 1, which is most useful when $N_1 \ge M$ and $M \ge 1$. The analysis for scheme 1 is similar to [9, 13], and we include the details below for the sake of completeness. We again divide the whole file set into two subsets $\mathcal{F}_1 = \{F_1, F_2, ..., F_{N_1}\}$ and $\mathcal{F}_2 = \{F_{N_1+1}, F_{N_1+2}, ..., F_N\}$. The files in $\mathcal{F}_1$ are the "more popular" files whose popularity is larger than $\frac{1}{KM_r}$. Recall that in our model each file is of unit length. The minimum indivisible portion of a file is called a "bit". We have assumed that each file has $|F|$ such bits. The cache placement strategy is given as follows.

Algorithm 1 Cache Placement Procedure

```
for $1 \le k \le K$, $1 \le n \le N_1$
    User $k$ randomly caches $\min\left(\frac{M|F|}{N_1}, |F|\right)$ bits of the file $F_n$
end for
```

Note that we only cache fractions of the N_1 popular files in the users' storage. On the other hand, the files requested by the K users may also come from files in $\mathcal{F}_2$. Assume that there are K_4 users requesting files in $\mathcal{F}_1$ and denote these users as U_1. Denote the other $K - K_4$ users requesting files in $\mathcal{F}_2$ as U_2. For every S that is a subset of U_1 and for every $k \in S$, let $V_{k,S\backslash\{k\}}$ represent all the bits that are requested by user k, that are stored in the cache of every other user of S except user k, and that are not stored in the caches of any other user in $U_1 \backslash S$. Denote $\oplus_{k\in S} V_{k,S\backslash\{k\}}$ as the XOR across the sets of bits $V_{k,S\backslash\{k\}}$. More precisely, order the bits in each $V_{k,S\backslash\{k\}}$ in some way. Then, each bit of $\oplus_{k\in S} V_{k,S\backslash\{k\}}$ is the XOR of the corresponding bits across $V_{k,S\backslash\{k\}}, k \in S$. Note that the size of $\oplus_{k\in S} V_{k,S\backslash\{k\}}$ equals to $\max\{|V_{k,S\backslash\{k\}}|, k \in S\}$.

Now we are ready to present the transmission scheme, which econsists of two steps. In the first step, the server will send coded data (as in the decentralized coded caching scheme of [8]) to meet the requests of users in U_1. In the second step, the server sends uncoded data to meet the requests of users in U_2. Recall that the size of U_1 is K_4.

After both steps, all requests of K users will be satisfied. The reason is as follows. If a user is in U_2, its request will be immediately satisfied in step 2. If a user k is in U_1, a bit b of its requested file will be in some $V_{k,S_b\backslash\{k\}}$, for a specific set S_b. After step 1, $V_{k,S_b\backslash\{k\}}$ will be retrieved by user k from the transmission received and its local storage. Hence, user K must be able to decode the bit b.

Algorithm 2 Transmission Procedure

```
Step 1: for $s = K_4, K_4 - 1, ..., 1$
            for every $S \subset U_1$ such that $|S| = s$, do
                Server sends $\oplus_{k\in S} V_{k,S\backslash\{k\}}$
            end for
        end for
Step 2: for every user $k \in U_2$
            Sever sends its requested file $d_k$
        end for
```

We now compute the rate required by the transmission scheme. This analysis is similar to [8]. We first calculate the rate R_1 sent by the server in step 1. For a subset $S \subset U_1$ and $|S| = s$, a bit of file d_k is in $V_{k,S\backslash\{k\}}$ with probability

$$(\frac{M}{N_1})^{s-1}(1 - \frac{M}{N_1})^{K_4-s+1}. \tag{42}$$

The expected number of bits in $V_{k,S\backslash\{k\}}$ is $|F| \cdot (\frac{M}{N_1})^{s-1}(1 - \frac{M}{N_1})^{K_4-s+1}$. When the file size F is large, the number of bits in $V_{k,S\backslash\{k\}}$ is $|F| \cdot (\frac{M}{N_1})^{s-1}(1 - \frac{M}{N_1})^{K_4-s+1} + o(|F|)$ with high probability. Therefore, the rate needed to be sent for a specific subset S is

$$\begin{aligned} |\oplus_{k\in S} V_{k,S\backslash\{k\}}| &= \max_{k\in S} |V_{k,S\backslash\{k\}}| \\ &= |F| \cdot (\frac{M}{N_1})^{s-1}(1 - \frac{M}{N_1})^{K_4-s+1} + o(|F|). \end{aligned} \tag{43}$$

In the sequel, we focus on the "large file-size" regime and ignore the factor $o(|F|)$. For each s, there are $C_{K_4}^s$ subsets S that satisfies $S \subset U_1$ and $|S| = s$. Summing over all possible s and all subsets S, the rate needed in the first step (in the unit of "bit") can be bounded by

$$\begin{aligned} R_1 &\le \sum_{s=1}^{K_4} C_{K_4}^s \cdot |F| \left(\frac{M}{N_1}\right)^{s-1} \left(1 - \frac{M}{N_1}\right)^{K_4-s+1} \\ &= |F|(1 - \frac{M}{N_1}) \frac{1 - (1 - \frac{M}{N_1})^{K_4}}{\frac{M}{N_1}} \\ &< |F|(\frac{N_1}{M} - 1). \end{aligned} \tag{44}$$

Note that this bound does not depend on K_4.

Next, we calculate the rate needed for step 2. Since each user requests a file in $\mathcal{F}_2$ with probability $\sum_{i=N_1+1}^N p_i$, the expected rate that it needs in step 2 is $F \sum_{i=N_1+1}^N p_i$. Summing over all K users, the expected rate of R_2 (again in the unit of "bit") can be represented by

$$R_2 \le K|F| \sum_{i=N_1+1}^N p_i. \tag{45}$$

Combining (44) and (45), and by a conversion from the unit of "bit" to the unit of "file" (recall that each file is unit length), we obtain that, the average transmission rate of scheme 1 must satisfy

$$R \le \frac{[N_1 - M]_+}{M} + K \sum_{i=N_1+1}^{N} p_i. \tag{46}$$

We now consider scheme 2, which is most useful when $N_1 \ge M$ and $M < 1$. In this case, we simply cache a fraction of M of file F_1, and broadcast uncoded files to satisfy users' requests. For the N_1 popular files, the rate needed would be no larger than $[N_1 - M]_+$. For the remaining files, the rate needed would be no larger than $\sum_{i>N_1} Kp_i$. Therefore, the average transmission rate of scheme 2 satisfies

$$R \le [N_1 - M]_+ + \sum_{i>N_1} Kp_i. \tag{47}$$

Note that (47) corresponds to the first term of (41) when $M < 1$.

Finally, we consider scheme 3, which is most useful when N_1 is smaller than M. Note that since $N_1 < M$, we can cache more files than N_1. Specifically, we use the following uncoded scheme. First, each user caches the $\lfloor M \rfloor$ most popular files. Then, if there are still some space left at each user (of the amount $M - \lfloor M \rfloor$), we can use it to cache part of file F_{N_3}. The rest of the files are not cached at all. Using only uncoded transmissions, it is easy to show that the average achievable rate of scheme 3 can be bounded as

$$R \le Kp_{N_3}(N_3 - M) + \sum_{i>N_3} Kp_i. \tag{48}$$

Note that both scheme 2 and scheme 3 correspond to the traditional LFU scheme with only uncoded transmission. Finally, for our achievable scheme, we choose the one (from the above three schemes) that has the lowest right-hand-side among (46), (47) and (48). Thus, the average transmission rate must satisfy (41).

VI. Comparison between the achievable rate and its lower bound

In this section, we will compare the achievable rate with its lower bound. Further, this analysis will reveal which one of the three schemes in Section V will be order-optimal in different cases. Recall that the achievable rate can be bounded as

$$\begin{aligned} R(K,\mathcal{F},\mathcal{P}) \le \min\Bigg(&\frac{[N_1 - M]_+}{\max(1,M)} + \sum_{i>N_1} Kp_i, \\ &Kp_{N_3}(N_3 - M) + \sum_{i>N_3} Kp_i \Bigg), \end{aligned} \tag{49}$$

where $N_3 = \lfloor M \rfloor + 1$, while the lower bound for all schemes is

$$\begin{aligned} R(K,\mathcal{F},\mathcal{P}) \ge \frac{1}{11} \max \Bigg\{ &\frac{1}{M_r}(N_1 + N_2 - M), \\ &\max_{N_x \ge N_1+1} Kp_{N_x}[N_x + N_y(N_x) - M] \Bigg\}, \end{aligned} \tag{50}$$

where $N_y(N_x) \triangleq \left\lfloor \frac{\sum_{i>N_x} p_i}{2p_{N_x}} + \frac{1}{2} \right\rfloor$.

Lemma 11: For all $N_x \ge N_1$, we have

$$1 + N_y(N_x) \ge \frac{\sum_{i\ge N_x} p_i}{4p_{N_x}}. \tag{51}$$

Proof: When $\sum_{i>N_x} p_i < p_{N_x}$, we have $N_y = 0$ and $1 + N_y \ge \frac{\sum_{i\ge N_x} p_i}{4p_{N_x}}$. The result of this lemma holds.

When $\sum_{i>N_x} p_i \ge p_{N_x}$, we have $N_y \ge 1$ and

$$\begin{aligned} 1 + N_y &\ge 1 + \frac{1}{2}(1 + N_y) \\ &\ge 1 + \frac{1}{2} \cdot \frac{\sum_{i>N_x} p_i}{2p_{N_x}} \\ &> \frac{\sum_{i\ge N_x} p_i}{4p_{N_x}}. \end{aligned} \tag{52}$$

Therefore, this lemma holds. ■

We divide the rest of proofs into 3 cases. We first consider case A when $N_1 \ge M$ and when the storage size satisfies $M \ge 3$. By definition, we have $M = M_r$. From the first term of (6), we have

$$R_{lb} \ge \frac{1}{11M}(N_1 + N_2 - M) \ge \frac{1}{11M}(N_1 - M). \tag{53}$$

Let $N_x = N_1 + 1$. Using the second term of the lower bound (6), we have

$$\begin{aligned} R_{lb} &\ge \frac{1}{11} Kp_{N_1+1}[N_1 + 1 + N_y(N_1+1) - M] \\ &\ge \frac{1}{11} Kp_{N_1+1}[1 + N_y(N_1+1)] \\ &\ge \frac{1}{44} K \sum_{i\ge N_1+1} p_i, \end{aligned} \tag{54}$$

where we have used Lemma 11 in the last inequality. On the other hand, the achievable rate can be bounded as

$$\begin{aligned} R_{up} &\le \min\left(\frac{1}{M}, 1\right) \cdot [N_1 - M]_+ + \sum_{i>N_1} Kp_i \\ &\le 11R_{lb} + 44R_{lb} \quad \text{(using (53) and (54))} \\ &= 55R_{lb}. \end{aligned} \tag{55}$$

We then consider case B when $N_1 \ge M$ and the storage size satisfies $M < 3$. Now $M_r = 3$ and $M < M_r$. From the first term of the lower bound (6), we have

$$R_{lb} \ge \frac{1}{11M_r}(N_1 + N_2 - M) = \frac{1}{33}(N_1 + N_2 - M) \ge \frac{1}{33} N_2. \tag{56}$$

Hence,

$$N_2 \le 33R_{lb}. \tag{57}$$

Note that the expression $\frac{1}{33}(N_1 + N_2 - M)$ in (53) is considerably lower than (56) when M is small. Thus, if we follow the same argument as in case A, the multiplicative gap will be poorer. Instead, the following proof uses a different strategy. Specifically, (57) implies that the sum popularity of all unpopular files cannot be very large, which can then be used to bound the contribution by the unpopular files.

Towards this end, we calculate the gap depending on the value of $\sum_{i\ge N_1+1} p_i$. When $\sum_{i\ge N_1+1} p_i \ge \frac{1}{KM_r}$, we have

$N_2 \geq 1$, and hence

$$R_{lb} \geq \frac{1}{33}(N_1 + 1 - M). \tag{58}$$

We further have, from the definition of N_2,

$$1 + N_2 \geq \frac{\sum_{i \geq N_1+1} p_i}{2/(KM_r)}. \tag{59}$$

Therefore, from the first term of the upper bound (50), the achievable rate can be bounded as

$$\begin{aligned} R_{up} &\leq (N_1 - M) + \sum_{i > N_1} Kp_i \\ &\leq 33R_{lb} - 1 + \frac{2(1+N_2)}{KM_r} \quad \text{(using (58) and (59))} \\ &\leq 33R_{lb} + \frac{2N_2}{M_r} \quad \left(\text{using } \frac{2}{KM_r} = \frac{2}{3K} < 1 \text{ and } \frac{1}{K} \leq 1\right) \\ &\leq 55R_{lb} \quad \text{(using } M_r = 3 \text{ and (57))}. \end{aligned} \tag{60}$$

Note that this bound corresponds to uncoded schemes, see (47).

When $\sum_{i \geq N_1+1} p_i < \frac{1}{KM_r}$, we will merge all the files whose index is greater than N_1 into one virtual file with popularity $p_v = \sum_{i \geq N_1+1} p_i$. Define another system with the set of files given by $\mathcal{F}_6 = \{V_0, F_1, ..., F_{N_1}, V\}$ and the corresponding popularity distribution as $\mathcal{P}_6 = \{v_0, \frac{1}{KM_r}, ..., \frac{1}{KM_r}, p_v\}$, where $v_0 = 1 - \frac{N_1}{KM_r} - p_v$. We then further decrease the popularity of files F_1,..., F_{N_1} to p_v. Let $\mathcal{P}_7 = \{1 - (N_1+1)p_v, p_v, ..., p_v, p_v\}$. Since only merging and reduction techniques are used in previous steps, we have

$$R(K, \mathcal{F}, \mathcal{P}) \geq R(K, \mathcal{F}_6, \mathcal{P}_6) \geq R(K, \mathcal{F}_6, \mathcal{P}_7). \tag{61}$$

We then apply the lower bound (49) to System $(K, \mathcal{F}_6, \mathcal{P}_7)$. From the second term of the lower bound (6),

$$\begin{aligned} R_{lb} &\geq \frac{1}{11} Kp_v(N_1 + 1 - M) \\ &\geq \frac{1}{11} \sum_{i \geq N_1+1} Kp_i. \end{aligned} \tag{62}$$

We further have, from the first term of the lower bound (6),

$$R_{lb} \geq \frac{1}{11M_r}(N_1 - M) = \frac{1}{33}(N_1 - M). \tag{63}$$

Therefore, from the first term of the upper bound (50), the achievable rate can be bounded as (again using uncoding scheme, see (47))

$$\begin{aligned} R_{up} &\leq (N_1 - M) + \sum_{i > N_1} Kp_i \\ &\leq 33R_{lb} + 11R_{lb} \\ &= 44R_{lb}. \end{aligned} \tag{64}$$

Finally, we consider case C when $N_1 < M$. In this case, we have $p_{N_3} \leq \frac{1}{KM_r}$. First, let $N_x = N_3$. Using the second term of the lower bound (6), we have

$$R_{lb} \geq \frac{1}{11} Kp_{N_3}(N_3 + N_y(N_3) - M) \geq \frac{1}{11} Kp_{N_3}(N_3 - M). \tag{65}$$

Then, we let $N_x = N_3 + 1$. Using the second term of the lower bound (6) again, we have

$$\begin{aligned} R_{lb} &\geq \frac{1}{11} Kp_{N_x}(N_x + N_y(N_x) - M) \\ &\geq \frac{1}{11} Kp_{N_3+1}(N_3 + 1 + N_y(N_3+1) - M) \\ &\geq \frac{1}{44} K \sum_{i > N_3} p_i, \end{aligned} \tag{66}$$

where we have used Lemma 11 in the last equality. On the other hand, with the second term of (50), the achievable rate can be bounded by

$$\begin{aligned} R_{up} &\leq Kp_{N_3}(N_3 - M) + \sum_{i > N_3} Kp_i \\ &\leq 55R_{lb} \quad \text{(using (65) \& (66))}. \end{aligned} \tag{67}$$

Combing all the cases above, we conclude that the gap is bounded by 55.

VII. Simulation Results

We next present numerical results that demonstrate the superior performance of the proposed scheme and discuss the insights from the results.

We will compare with four other schemes. The first one is an uncoded scheme, i.e., least-frequently used (LFU) caching strategy [18], which caches the M most popular files in all users' storage. The second scheme is the decentralized uniform coded caching scheme in [8], where a $\frac{M}{N}$ portion of every file is cached in each user's storage, regardless of its popularity. The remaining two are group-caching [9] and RLFU [13], both of which consider heterogenous popularity. In group-caching [9], files with close popularity (differing by at most a factor of 2) are grouped together. The scheme in [9] assigns an equal fraction of the cache space to each group, and performs coded transmission only among users requesting files from the same group. The rate needed for each group is calculated according to Equation (1) in [9], and the overall rate is the summation of the rate for each group. In our simulation, we further allow group-caching to optimize the allocation of cache space for each group to minimize this overall rate, which will likely enhance its performance. However, as we will see shortly, the requirement that coded transmission is only performed among users requesting files from the same group still becomes a limiting factor, and as a result the performance of group-caching can be even worse than uniform coded-caching.

Finally, as we discussed earlier, the RLFU scheme proposed in [13] is similar to ours in that they both evenly cache files whose popularity is above a threshold. Note that [13] proposed a way to choose the optimal threshold for RLFU based on a 1-dimensional optimization over all possible threshold values. When $N_1 \geq M$ and $M \geq 3$, since our proposed algorithm (using the simple threshold N_1) can also be viewed as a member in the class of RLFU algorithms, it implies that RLFU with the optimal threshold will achieve better performance. What is interesting, however, is that in most of the simulation settings below, our simple choice N_1 performs almost as well as RLFU with the optimal threshold, which suggests that the simple choice N_1 is in fact quite close-to-optimal.

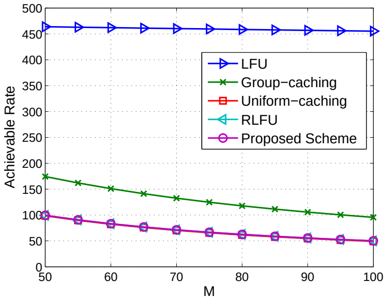


Fig. 2: N=5000,K=500,α=0.2.

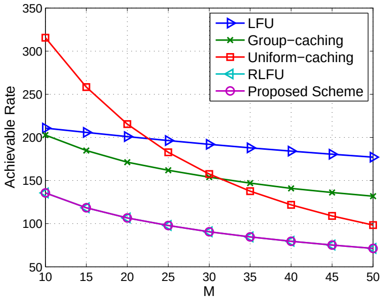


Fig. 3: N=5000,K=500,α=1.1.

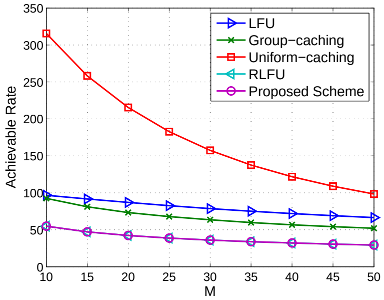


Fig. 4: N=5000,K=500,α=1.4.

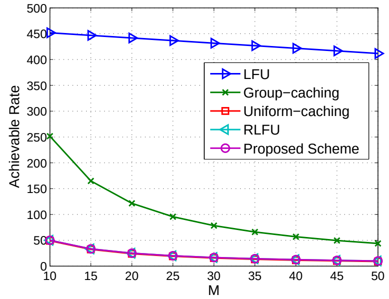


Fig. 5: N=500,K=5000,α=1.

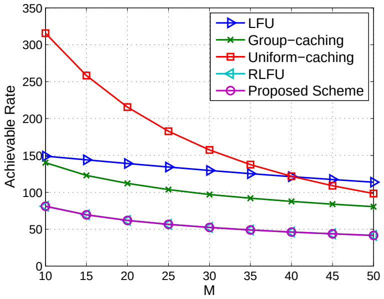


Fig. 6: N=5000,K=500,α=1.4, r=2.

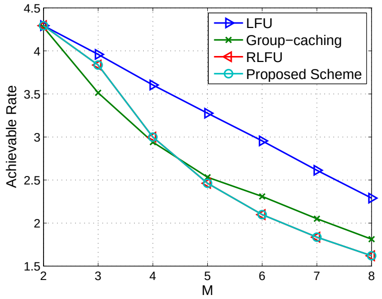


Fig. 7: N=21,K=12,step function popularity.

After the cache placement setting, we simulate the request and transmission processes. The requests of all users are generated randomly according to the file popularity distribution. The rate of each scheme is calculated as follows. 1) For files that are not cached at all but are requested, the rate is calculated as the distinct number of such files. 2) For files that are cached, the rate is calculated as follows, depending on the scheme. For LFU, a cached file is always cached in its entirety. Thus, the rate is zero. For other schemes, a file may be partially cached. Specifically, the rate for uniform caching is $(1-\frac{M}{N})(1-(1-\frac{M}{N})^K)/(\frac{M}{N})$, similar to (44). The rate for group caching is the summation of the rates for each group, with uniform caching applied within each group. The rate for RLFU and our scheme is calculated by applying uniform caching to only the popular files. The threshold for popular files in RLFU is optimized according to Eqs. (16) and (17) in [13].

In the following figures, we present the mean transmission rate calculated from a large number ($>$ 1000) of request patterns randomly generated according to the popularity distribution. The confidence intervals are very small and thus not shown.

Comparison under Zipf popularity distribution: The first set of numerical results are for Zipf popularity distribution, i.e., the popularity of the i-th popular file is $p_i = \frac{H(\alpha)}{i^\alpha}$, where α is the Zipf exponent and $H(\alpha)$ is the normalization factor. Note that $\alpha > 1$ means that the distribution is heavily skewed to the most popular files, while $\alpha < 1$ means that the distribution is “flatter”. We simulate a system with $K = 500$ users and $N = 5000$ files. In Figures 2-4, we plot the achievable transmission rate as a function of users’ storage size M. As readers can see, our proposed scheme achieves the best performance under all scenarios. Specifically, in Figure 2, $\alpha = 0.2$ is small, which implies that files have a “flat” popularity distribution. We can observe that LFU performs poorly because it doesn’t exploit coded transmission opportunities. A deeper investigation reveals that both RLFU and our scheme turn into uniform coded caching, which outperform group-caching. In contrast, in Figure 4, $\alpha = 1.4$ is large, which implies that a small fraction of files dominate the popularity distribution. Uniform coded-caching performs poorly because it neglects the significant popularity difference. On the other hand, by preferably caching the most popular files, both LFU, RLFU, group-caching and our scheme all perform well.

Finally, from Figures 2-4, we can see that group-caching appears to also exhibit robust performance, except in Figure 5. On the other hand, our scheme and RLFU consistently performs better. We do emphasize that the simulation of both RLFU and group-caching involves an extra step of optimizing cache allocation across groups. In contrast, our scheme is much simpler and does not involve such an additional optimization step. Thus, our proposed scheme not only achieves better performance, but also is easy to implement.

Comparison under non-Zipf distribution: Next, we simulate these algorithms under a non-Zipf distribution, i.e., Zipf-Mandelbrot law distribution, where the i-th popular file is requested with probability $p_i = \frac{H(\alpha)}{(i+r)^\alpha}$ for a constant $r = 2$ and $\alpha = 1.4$. The simulation results are presented in Figure 6. Again, our scheme and RLFU achieve the best performance.

As we mentioned earlier, in most of the simulations (Figs. 2-6), both our proposed scheme and RLFU [13] achieve very similar performance, and both are better than all other schemes simulated. However, there are scenarios where other schemes, e.g., group caching, perform better. An example is shown in Fig. 7. In Fig. 7, there are one file with popularity of $\frac{5}{9}$, 10 files with popularity of $\frac{1}{30}$ and 10 files with popularity of $\frac{1}{90}$. We look into the $M = 3$ case. If we focus on the case when $M = 3$, a closer inspection indicates that both RLFU and our proposed scheme turns into LFU, i.e., the most popular $M = 3$ files are cached in their entirety. What is interesting is that group-caching turns out to out-perform RLFU. For this setting, it turns out that group caching scheme caches the first file in their entirety, caches the following 10 files evenly using the remaining cache size, and does not cache the rest of the 10 files (with popularity $\frac{1}{90}$). In general, this simulation result thus suggests that, in some cases, it may be useful to use 2 thresholds, instead of using 1 threshold as in our scheme and RLFU. We leave the design of such 2-threshold scheme as a topic for future work.

VIII. Conclusion

In this work, given an arbitrary popularity distribution, we first derive a new information-theoretical lower bound on the expected transmission rate of any coded caching schemes. We then show that a simple coded-caching scheme attains an expected transmission rate that is at most a constant factor away from the lower bound. Unlike other existing studies, the constant factor that we derived is independent of the popularity distribution.

There are a number of interesting questions for future studies. First, the complexity of the transmission scheme in Section V can be high (esp. for enumerating all the subsets S). Thus, an important question is whether we can develop low-complexity transmission schemes that still attain similar performance guarantees. Further, it would be interesting to study how the benefits of coded caching can be extended to wireless environments (in particular heterogeneous wireless networks).

Acknowledgment

This work was partially supported by NSF grants CNS-1457137 and CNS-1703014, a grant from the Army Research Office W911NF-14-1-0368, and grants from NSF China (No. 61532012, No. 61325012, No. 61601287).

References

[1] J. Zhang, X. Lin and X. Wang, “Coded Caching under Arbitrary Popularity Distributions”, in *Information Theory and Applications Workshop*, UCSD, USA, Feb. 2015.

[2] White Paper, “Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2013-2018”, Feb. 2014.

[3] D. Wessels, *Web Caching*. O’Reilly, 2001.

[4] N. Golrezaei, K. Shanmugam, A.G. Dimakis, A.F. Molisch and G. Caire, “FemtoCaching: Wireless Video Content Delivery through Distributed Caching Helpers”, in *Proc. IEEE INFOCOM*, Orlando, USA, Mar. 2012.

[5] V. Shah and G. de Veciana, “Performance Evaluation and Asymptotics for Content Delivery Networks”, in *Proc. IEEE INFOCOM*, Toronto, Canada, Apr. 2014.

[6] B. Tan and L. Massoulie, “Optimal Content Placement for Peer-to-Peer Video-on-Demand Systems”, *IEEE/ACM Trans. Netw.*, vol. 21, no. 2, pp. 566-579, Apr. 2013.

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

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

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

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

[11] M. Ji, A. Tulino, J. Llorca and G. Caire, “On the Average Performance of Caching and Coded Multicasting with Random Demands”, in *11th International Symposium on Wireless Communication Systems (ISWCS),* Barcelona, Spain, Aug. 2014, pp. 922-926.

[12] M. Ji, A. Tulino, J. Llorca and G. Caire, “Order Optimal Coded Caching-Aided Multicast under Zipf Demand Distributions”, *arXiv:1402.4576v1 [cs.IT]*, Feb. 2014.

[13] M. Ji, A. Tulino, J. Llorca and G. Caire, “Order-Optimal Rate of Caching and Coded Multicasting with Random Demands”, *arXiv:1502.03124v1 [cs.IT]*, Feb. 2015.

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

[15] M. Ji, A. Tulino, J. Llorca and G. Caire, “Order Optimal Coded Delivery and Caching: Multiple Groupcast Index Coding”, *arXiv:1402.4572 [cs.IT]*, Feb. 2014.

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

[17] R. Kaas and J.M. Buhrman, “Mean, Median and Mode in Binomial Distributions”, *Statistica Neerlandica*, vol. 34, no. 1, pp. 13-18, Mar. 1980.

[18] D. Lee, S. H. Noh, S. L. Min, J. Choi, J. H. Kim, Y. K. Cho and C. S. Kim, “LRFU: A Spectrum of Policies that Subsumes the Least Recently Used and Least Frequently Used Policies”, *IEEE Trans. Computers*, vol. 50, no. 12, pp. 1352-1361, 2001.

[19] Hermann Thorisson, *Coupling, Stationarity, and Regeneration*. Springer, 2000.

Appendix

A. *Proof of Lemma 2*

Proof: For a given cache placement and transmission scheme $\mathfrak{F}$, let $r = r_{\mathfrak{F}}(W_i)$ be the transmission rate for the random request pattern W_i in $\mathbb{W}(K, \mathcal{F}_1, \mathcal{P}_1)$, where W_i is chosen randomly in $\mathbb{W}(K, \mathcal{F}_1, \mathcal{P}_1)$. Note that since W_i is random, r is also a random variable. Further, the average transmission rate is $R_{\mathfrak{F}}(K, \mathcal{F}, \mathcal{P}) = E[r]$. Similarly, let $r' = r_{\mathfrak{F}}(W_j)$ denote the transmission rate, where W_j is chosen randomly from the new distribution in $\mathbb{W}(K, \mathcal{F}_1, \mathcal{P}_2)$. Again, $R_{\mathfrak{F}}(K, \mathcal{F}_1, \mathcal{P}_2) = E[r']$. Note that the two expectations corresponding to two distributions, driven by $\mathbb{W}(K, \mathcal{F}_1, \mathcal{P}_1)$ and $\mathbb{W}(K, \mathcal{F}_1, \mathcal{P}_2)$, respectively. Therefore, r and r' cannot be directly compared in a pointwise manner, e.g., the request probabilities for a same pattern are different.

In the following, we will show that $r' \leq^D r$, i.e., r' is stochastic dominated by r. To do so, we will use Theorem 3.1 in [19]. Specifically, we need to find two coupled variables $\hat{r}$ and $\widehat{r'}$ with the following properties:

(a) r and $\hat{r}$ have the same distribution;

(b) r' and $\widehat{r'}$ have the same distribution;

(c) $\hat{r} \geq \widehat{r'}$ almost surely.

Then, applying Theorem 3.1 in [19], we can conclude that $r' \leq^D r$ and $E(r) = E(\hat{r}) \geq E(\widehat{r'}) = E(r')$.

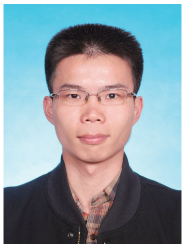
It remains to show the existence of $\hat{r}$ and $\hat{r'}$, which are constructed as follows. For every $W_i \in \mathbb{W}(K, \mathcal{F}_1, \mathcal{P}_1)$, we let $\hat{r} = r$ and thus they must have exactly the same distribution, satisfying property (a). We then map W_i to another request W_i'. For every non-empty file F_j in W_i requested by a user, with probability $\frac{1/(KM_r)}{p_j}$ we keep it in $W_i^{'}$. With probability $1 - \frac{1/(KM_r)}{p_j}$, we replace it by the empty file F_0.

By such a construction, we now verify property (b) holds. We wish to show that the probability with which the mapped request patterns is $W_i' = \{f_1, f_2, ..., f_K\}$, where f_k is the file requested by user k, is the same as the probability with which the same pattern $W_j = \{f_1, f_2, ..., f_K\}$ is requested in $\mathbb{W}(K, \mathcal{F}_1, \mathcal{P}_2)$. We first fix a user k and compare the probability that user k requests file f_k. If $f_k \in \{F_1, F_2, ..., F_{N_1}\}$, the probability $p_{W_i'}(f_k)$ that user k requests f_k in the mapped request pattern is equal to the probability that user k requests f_k in $\mathbb{W}(K, \mathcal{F}_1, \mathcal{P}_1)$, conditioned on which the request file f_k is not mapped to the empty file during the mapping. Thus, $p_{W_i'}(f_k) = p(f_k) \cdot \frac{1/(KM_r)}{p(f_k)} = \frac{1}{KM_r}$, which is the same as the probability $p_{W_j}(f_k)$ that user k requests f_k in $\mathbb{W}(K, \mathcal{F}_1, \mathcal{P}_2)$. If $f_k = \emptyset$, according to our construction, the probability $p_{W_i'}(f_k)$ that user k requests f_k in the mapped request pattern is equal to $1 - \sum_{k=1}^{N_1} p_{W_i'}(f_k) = 1 - \frac{N_1}{KM_r}$. Hence, it is also the same as the probability $p_{W_j}(f_k)$ that user k requests the empty file in $\mathbb{W}(K, \mathcal{F}_1, \mathcal{P}_2)$. Finally, the probability that the mapped request pattern is W_i', is

$$\begin{aligned} P(W_i' = \{f_1, ..., f_K\}) &= \prod_{k=1}^{K} p_{W_i'}(f_k) \\ &= \prod_{k=1}^{K} p_{W_j}(f_k) \\ &= P(W_j = \{f_1, ..., f_K\}). \end{aligned}$$

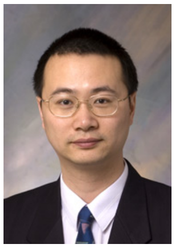
Thus, W_i' has the same distribution as W_j, and hence $\hat{r'}$ and r' have the same distribution, satisfying property (b).

Further, for any caching and transmission scheme that can satisfy users' request W_i, it must can satisfy W_i', since W_i' can be seen as a subset of W_i. Hence, the rate to serve the request pattern W_i' is clearly no larger than the rate to serve the request pattern W_i. Thus we have $\hat{r} \geq \hat{r'}$ almost surely, satisfying property (c). The result of the lemma then follows. ■



Jinbei Zhang received the B.S. degree in Electronic Engineering from Xidian University, Xi'an, China, in 2010, and the Ph.D. degree in electronic engineering at Shanghai Jiao Tong University, Shanghai, China, in 2016.

His current research interests include network security, network virtualization and caching.



Xiaojun Lin (S'02 M'05 SM'12 F'17) received his B.S. from Zhongshan University, Guangzhou, China, in 1994, and his M.S. and Ph.D. degrees from Purdue University, West Lafayette, IN, in 2000 and 2005, respectively. He is currently a Professor of Electrical and Computer Engineering at Purdue University.

Dr. Lin's research interests are in the analysis, control and optimization of large and complex networked systems, including both communication networks and power grid. He received the IEEE INFOCOM 2008 best paper and 2005 best paper of the year award from Journal of Communications and Networks. His paper was also one of two runner-up papers for the best-paper award at IEEE INFOCOM 2005. He received the NSF CAREER award in 2007. He was the Workshop co-chair for IEEE GLOBECOM 2007, the Panel co-chair for WICON 2008, the TPC co-chair for ACM MobiHoc 2009, and the Mini-Conference co-chair for IEEE INFOCOM 2012. He is currently serving as an Area Editor for (Elsevier) Computer Networks Journal, and has served as an Associate Editor for IEEE/ACM Transactions on Networking and a Guest Editor for (Elsevier) Ad Hoc Networks journal.

Xinbing Wang received the B.S. degree (with hons.) in automation from Shanghai Jiao Tong University, Shanghai, China, in 1998, the M.S. degree in computer science and technology from Tsinghua University, Beijing, China, in 2001, and the Ph.D. degree with a major in electrical and computer engineering and minor in mathematics from North Carolina State University, Raleigh, in 2006. Currently, he is a Professor in the Department of Electronic Engineering, and Department of Computer Science, Shanghai Jiao Tong University, Shanghai, China. Dr. Wang has been an Associate Editor for IEEE/ACM TRANSACTIONS ON NETWORKING, IEEE TRANSACTIONS ON MOBILE COMPUTING, and ACM Transactions on Sensor Networks. He has also been the Technical Program Committees of several conferences including ACM MobiCom 2012,2014, ACM MobiHoc 2012-2017, IEEE INFOCOM 2009-2017.