Zhang, C., & Peleato, B. (07-11 June 2020). On the Average Rate for Coded Caching with Heterogeneous User Profiles [proceedings]. ICC 2020: IEEE International Conference on Communications. Virtual Conference.

DOI: https://doi.org/10.1109/icc40277.2020.9148779

On the Average Rate for Coded Caching with Heterogeneous User Profiles

Ciyuan Zhang and Borja Peleato,
Purdue University
West Lafayette, IN 47907
Email:{zhan3375,bpeleato}@purdue.edu

***Abstract*—Coded caching utilizes pre-fetching during off-peak hours and multi-casting for delivery in order to balance the traffic load in communication networks. Most of the existing research focuses on reducing the peak transmission rates with homogeneous file popularities, despite modern systems are often able to categorize users by their preferences and tend to care more about the average rather than peak rate. This paper considers a scenario with heterogeneous user profiles and analyzes the average transmission rates for three coded caching schemes under the assumption that each user can only request a subset of the total available files. In addition, it evaluates the average rate of the three schemes when the number of files is much larger than the number of users and the amount of cache memory. Furthermore, it proposes methods of cache allocations which minimize the average rate when the users have relatively small storage. Our results demonstrate connections between cache distributions which result in minimal average rate and peak rate.**

I. Introduction

With the high demand for data driven by state of the art computer communication networks, coded caching was introduced to drastically reduce the congestion of the systems during their peak hours. In [1], Maddah-Ali and Niesen proposed a coded caching scheme which maximizes multicasting opportunities for the worst case user demands. Subsequent works focused on lowering the peak rate in different scenarios [2], [3]. However, in practical systems it is common for different users to request the same files, thus studying the expected rate over all possible user demand profiles can be more useful towards modeling and developing practical caching schemes. Some works have studied the expected rate, e.g. [4]–[6], but not with multiple heterogeneous users.

Papers like [7]–[9] have addressed the significance of predicting users searching behavior according to their preferences. This mirrors the current trend of online video streaming companies like Hulu and Netflix which spend a considerable amount of money investigating their customers' habits and categorizing them according to their streaming preferences. The paper [10] addressed a system where the users are grouped into classes with similar file interests. It proposed three coded caching schemes for this scenario and studied their peak rate. However, in practical systems, the peak rate is infrequently achieved and the average rate is a preferred representation of cost, speed, and overall performance.

In this paper, we aim to characterize the average rate of the three schemes proposed in [10] and compare them asymptotically. In practical computer communication networks, the number of files online is exceedingly larger than the number of users and their storage capacity. The streaming experience is remarkably affected by the shortage of local storage at the users. This paper provides suggestions for choosing a scheme in a system with heterogeneous user profiles where the number of users and quantity of memory is much smaller than the number of files. Furthermore, the heuristics in this paper provide users with instructions on how to allocate their limited cache memory.

This paper will be organized as follows: Section II – system model and schemes utilized, Section III - calculating average rate, Section IV - results, Section V – simulations, and Section VI - conclusion.

II. System model

The average rate of a system with heterogeneous users was addressed in [6], [11], [12], but just for the case of two users. This paper considers a system with a single server connected to K end users through an error-free broadcast link. The server is storing N files of F bits and the end users have independent cache storage of MF bits each. The N files are divided into common files, which may be requested by all users, and unique files, such as cartoon or sci-fi movies, which are only appealing to a subset of the users. The K users are divided into G classes according to the different types of files that they may request. In this paper, we assume that the number of users and unique files is the same for each class, and they do not overlap with each other. Therefore, the number of users per class is $\frac{K}{G}$ and $N = N_c + GN_u$, where N_c is the number of common files and N_u is the number of unique files per class. This scenario is illustrated in Figure 1 with only two classes.

A coded caching scheme contains a placement phase and a delivery phase. In the placement phase, the server first partitions all the files into segments of equal size and stores them into some users' cache. In the delivery phase each user makes a file request, the server delivers coded segments to satisfy users' requests. In this paper, we assume that every user requests a single file in the delivery phase to demonstrate the congested time of the network system.

This paper adapts the centralized coded caching scheme with uncoded prefetching proposed by Maddah-Ali and Niesen [1], from this point on referred to as MN's scheme, to heterogeneous user profiles. In the placement phase, MN's scheme splits each file into $\binom{K}{t}$ distinct segments, where

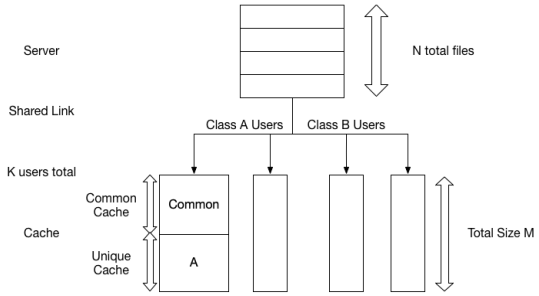

FIGURE 1: System Model with two distinct classes, A and B, each having 2 users. Each user's cache is divided into cache for common or unique files.

$t = \frac{KM}{N}$. Each segment is cached by a distinct set of t users, which results in each user caching $\binom{K-1}{t-1}$ segments per file. Specifically, this scheme can satisfy any vector of requests by transmitting at most $\binom{K}{t+1}$ messages of size $\binom{K}{t}$ bits. The peak rate (normalized by the file size F) is written as

$$R_{MN}(K,t) = \frac{\binom{K}{t+1}}{\binom{K}{t}} = \frac{K-t}{t+1}. \tag{1}$$

If only some of the users make a request and their requests overlap, or the server only receives requests for m files, then the transmission rate with MN's scheme will become

$$R(K,m,t) = \frac{\binom{K}{t+1} - \binom{K-m}{t+1}}{\binom{K}{t}}, \tag{2}$$

as was shown in [4].

In our simulations, we will introduce several approximations to extend Eq. (2) into a continuous function over $0 \leq t \leq K$, which we now describe.

When $t \leq 1$, the joint memory of the users is not enough to cache every file in full. In that case, there will be a portion of each file which is not cached anywhere, and therefore needs to be transmitted uncoded whenever any user requests that file. The rest of the file can be transmitted using coded caching schemes. By denoting the fraction of each file being left out as p, it can be written that $KM = (1-p)N$. Therefore, according to [13], the overall transmission rate when $t \leq 1$ is

$$R = N(\vec{d})p + [\text{Rate if } t = 1](1-p), \tag{3}$$

where $N(\vec{d})$ is the number of distinct files being requested.

When $K - 1 < t \leq K$, there will be a portion of each file which is cached by every user and thus never needs to be transmitted. We use coded caching to transmit the rest. Denoting the portion of file segment being cached by all the users as γ, the transmission rate when $t \in (K-1, K]$ is

$$R = 0 \cdot \gamma + [\text{Rate if } t = K-1](1-\gamma). \tag{4}$$

When $t > 1$ and is not an integer, Eq. (2) is not well defined because the combinatorial expressions require the coefficients to be strictly non-negative integers. Therefore, the Gamma function was used to extend their domain. Since the Gamma function satisfies $\Gamma(n) = (n-1)!$, the binomial coefficients in Eq. (2) can be interpolated continuously as

$$\binom{n}{k} = \frac{n!}{k!(n-k)!} = \frac{\Gamma(n+1)}{\Gamma(k+1)\Gamma(n-k+1)} \tag{5}$$

without any error when they are integers.

The schemes utilized in this paper are based on MN's scheme and were first proposed in [10].

- Scheme 1: the system behaves as if all files are common. It ignores the differences between all user profiles and requires every user to cache segments from every file, even if it would never request them.
- Scheme 2: the system decouples the caching and delivery of common and unique files. Each user devotes a fraction x of its cache to store segments from common files and $(1-x)$ to unique files. The placement and delivery of both common and unique files are done independently according to MN's scheme. This paper optimizes the ratio x so that a minimum average rate is achieved.
- Scheme 3: the system treats all files as unique files. It ignores the fact that some files can be requested by all user classes and independently applies MN's scheme for placement and delivery phases within each class of users.

III. Calculating Average Rate

In the delivery phase, each user k makes a request d_k to the server and we denote the probability mass function (pmf) of the random request d_k as $p_{d_k}^{[k]}$.

Definition 1. *The demand set for user k is defined as $S_k \triangleq \left\{ n \in [N_c + GN_u] : p_n^{[k]} > 0 \right\}$, which represents the set of distinct files that are demanded by user k with a positive probability p.*

Now we derive the average rate for the three schemes.

- Scheme 1: The average rate according to Eq. (2) is:

$$R_{\text{avg}}^{(1)} = \sum_{\forall \vec{d}} p_{\vec{d}} \frac{\binom{K}{t_1+1} - \binom{K-N_1(\vec{d})}{t_1+1}}{\binom{K}{t_1}}, \tag{6}$$

  where $N_1(\vec{d})$ denotes the number of distinct files requested by the K users and $t_1 = \frac{KM}{N_c + GN_u}$. This scheme behaves as if any user can demand any of the $N_c + GN_u$ files, despite each user only demands from a set of $N_c + N_u$ files corresponding to its class.
- Scheme 2: The average rate can be calculated as:

$$R_{\text{avg}}^{(2)} = R_c + \sum_{i=1}^{G} R_u, \tag{7}$$

  consisting of the average transmission rate for transmitting common files R_c and that for unique files R_u in each class. According to Eq. (2), R_c is given by

$$R_{\text{c}} = \sum_{\forall \vec{d}} p_{\vec{d}} \frac{\binom{K}{t_c+1} - \binom{K-N_c(\vec{d})}{t_c+1}}{\binom{K}{t_c}}, \tag{8}$$

where $N_c(\vec{d})$ denotes the number of distinct common files requested by the K users and $t_c = \frac{KMx}{N_c}$, with x denoting the ratio of storage which the user allocates to common file segments. Similarly, the average transmission rate for unique files in each class is written as

$$R_u = \sum_{\forall \vec{d}} p_{\vec{d}} \frac{\binom{\frac{K}{G}}{t_u+1} - \binom{\frac{K}{G}-N_{u_i}(\vec{d})}{t_u+1}}{\binom{\frac{K}{G}}{t_u}}, \tag{9}$$

where $t_u = \frac{KM(1-x)}{GN_u}$, and $N_{u_i}(\vec{d})$ denotes the number of distinct unique files requested by the K/G users in the i-th class.

- Scheme 3: The average rate when considering all files as unique is calculated by independently applying Eq. (2) to each class of users, resulting in:

$$R^{(3)}_{\text{avg}} = \sum_{i=1}^{G} \sum_{\forall \vec{d}} p_{\vec{d}} \frac{\binom{\frac{K}{G}}{t_3+1} - \binom{\frac{K}{G}-N_{3_i}(\vec{d})}{t_3+1}}{\binom{\frac{K}{G}}{t_3}}, \tag{10}$$

where $t_3 = \frac{KM}{G(N_c+N_u)}$ and $N_{3_i}(\vec{d})$ represents the number of distinct files requested by the K/G users in the i-th class.

Henceforth we will focus on analyzing Scheme 2 when K is very large, because as later proven, it is able to achieve a lower average rate than Schemes 1 and 3 in the asymptotic regime, *i.e.* , when $N_c = N_u$ and both are much larger than K and M. Now we derive approximations to Eq. (7).

To determine the cumulative distribution function (cdf) of a random request, we assume that the users' demands are independent and uniformly distributed over the demand set. When the number of users K is large enough, the number of distinct files being requested is approximated to follow a Gaussian distribution whose mean and variance can be derived using the method of indicators.

Lemma 1. *Under the previous assumptions, the expected number of distinct files requested by K users from N distinct files in the delivery phase of a coded caching system is*

$$\mathbb{E}[X] = N\Big[1 - \Big(\frac{N-1}{N}\Big)^K\Big]. \tag{11}$$

Proof. See Appendix. □

When the number of users and files is large, the number of distinct files being requested becomes large and the variance of the normal distribution is negligible. According to the law of large numbers, the number of distinct files requested should be close to the expected value. We will approximate the number of distinct files being requested as a constant, and approximate the variance as zero due to the large number of distinct files being requested.

In Scheme 2, the number of distinct unique files requested by each class of users can be approximated to be identical for all classes when the number of classes is much smaller than the number of unique files and users per class. With these approximations, Eq. (7) becomes

$$R^{(2)}_{\text{avg}} \simeq \frac{\binom{K}{t_c+1} - \binom{K-\mathbb{E}[N_c(\vec{d})]}{t_c+1}}{\binom{K}{t_c}} + G\frac{\binom{\frac{K}{G}}{t_u+1} - \binom{\frac{K}{G}-\mathbb{E}[N_u(\vec{d})]}{t_u+1}}{\binom{\frac{K}{G}}{t_u}}. \tag{12}$$

In this equation, $t_c = \frac{KMx}{N_c}$, $t_u = \frac{KM(1-x)}{GN_u}$, and according to Lemma 1,

$$\mathbb{E}[N_c(\vec{d})] = N_c\Big[1 - \Big(\frac{N_c+N_u-1}{N_c+N_u}\Big)^K\Big], \tag{13}$$

$$\mathbb{E}[N_u(\vec{d})] = N_u\Big[1 - \Big(\frac{N_c+N_u-1}{N_c+N_u}\Big)^{K/G}\Big]. \tag{14}$$

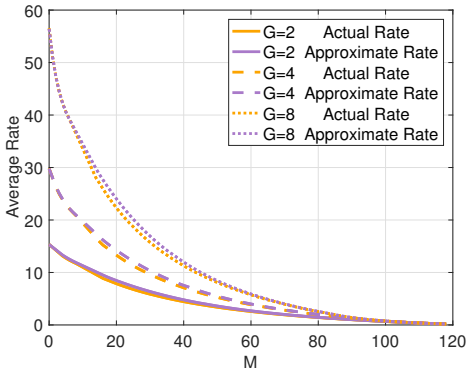

FIGURE 2: Actual rate vs Approximate rate for $N_c = 64$, $N_u = 64$, and 8 users per class.

Figure 2 compares the average rate from Eq. (7) and its approximation in Eq. (12) for a system with $N_c = 64$, $N_u = 64$ and 8 users per class. The approximation error is within 2 percent of the exact value, even when the number of files and users is small. Therefore, in Section IV we investigate Eq. (12) instead of the exact rate.

IV. Results

This section will attempt to optimize the distribution of cache between common and unique files in Scheme 2 (namely x) so that the average rate is minimal when the amount of memory is small. It will also compare the asymptotic average rate of the three schemes in Section III when $N_c = N_u$ and both are much larger than the number of users and their cache capacity. It will consider different regimes where the number of files of one type is excessively larger than that of the other, and propose an optimal cache partition to reach a minimal asymptotic average rate for Scheme 2.

First, we study how to optimally split the cache between common and unique files in Scheme 2 (i.e, optimizing x) when the local storage is relatively small.

Proposition 2. *When utilizing Scheme 2, given the conditions $N_c \geq \frac{K-1}{4K}GN_u$, the user should always devote all the cache to common files ($x = 1$) when $M \leq \frac{GN_u-2N_c}{2K+\frac{GN_u}{N_c}}$ so that the average rate is minimized.*

Proof. See Appendix. □

Corollary 1. *When the users' devices have relatively small storage compared to the amount of the files, and the number of common files is larger than a quarter of the total number of unique files, it is recommended for the users to devote all the cache to common files and transmit the unique files uncoded.*

The proposition above provides the optimal x when M is relatively small. When M is greater than the threshold given, we should decrease x to achieve the lowest average rate.

Now we turn our attention to asymptotic regimes where the number of files is remarkably larger than the number of users, as in a practical system.

Proposition 3. *Scheme 2 achieves a lower average rate than Schemes 1 and 3 when $N_c = N_u \to \infty$, provided K and M are large but much smaller than N_c (e.g. $K = M = \sqrt{N_c}$).*

Proof. See Appendix. □

Our simulation results, as well as the above proposition, suggest that the following hypothesis might be true, but we have been unable to prove it:

Hypothesis. *When $K < N_c, K < N_u$ and the users' local memory is very small, Scheme 2 achieves a lower average rate of transmission than 1 or 3.*

Furthermore, we extend the asymptotic regime to the situation when the number of common and unique files are not equal. We will show that the optimal x also depends on the comparison between N_c and N_u.

Proposition 4. *In Scheme 2, we assume that G and K are both much smaller than N_c and N_u, when $N_u << N_c \to \infty$, the users should always cache all the common files ($x = 1$) and transmit the unique files uncoded so that the average rate is minimal. Contrarily, when $N_c << N_u \to \infty$, the users should always devote all the cache to unique files ($x = 0$) to reach the minimal average rate.*

Proof. See Appendix. □

The specialty of Scheme 2 is that its flexibility to devote the users' cache to different files according to the parameters of the system. The Proposition 4 demonstrates that when the amount of files of one type is exceedingly larger than that of the other, the system should devote all of its cache to the dominant type of files, which aligns with our intuition.

V. Simulations

The minimal average rate of a coded caching system with heterogeneous user profiles is unknown. In order to find an approximate lower bound for the achievable rate, we define a new scheme "MN with oracle", where the system knows in advance whether each user will request a common or unique file and populates the caches accordingly. Using MN's placement and delivery scheme for common and unique files separately yields the following average rate for this scheme:

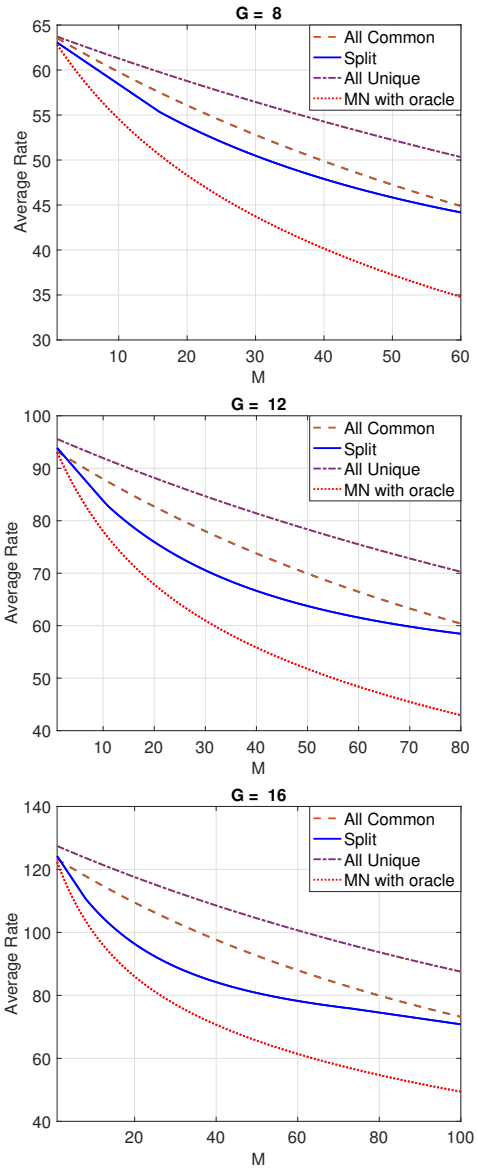

FIGURE 3: Actual rate vs Cache Size (M) for $N_c = 1024$, $N_u = 1024$, and 8 users per class.

$$R_{\text{orc}} = \sum_{K_c=0}^{K} p_{K_c}\Big[R(K_c, m_c, t_{oc}) + GR\Big(\frac{K-K_c}{G}, m_u, t_{ou}\Big)\Big], \tag{15}$$

where $R(K, m, t)$ is defined in Eq. (2), K_c represents the number of users that request common files, $t_{oc} = \frac{K_c M}{N_c}$, $t_{ou} = \frac{(K-K_c)M}{GN_u}$, m_c and m_u are the expected number of distinct common and unique files requested, respectively, and

$$p_{K_c} = \binom{K}{K_c}\Big(\frac{N_c}{N_c+N_u}\Big)^{K_c}\Big(\frac{N_u}{N_c+N_u}\Big)^{K-K_c}. \tag{16}$$

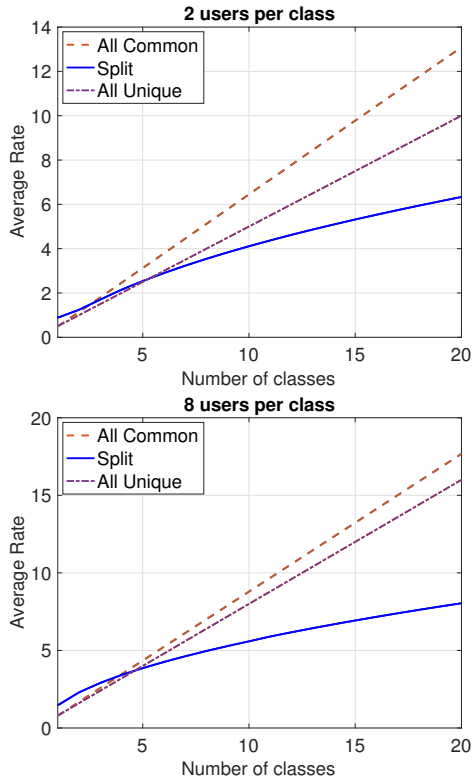

FIGURE 4: Average rate vs number of classes for $N_c = 64$, $N_u = 64$, and $M = 64$.

Figure 3 compares the average rate provided by the three schemes in Section III with this lower bound for a system with $N_c = 1024, N_u = 1024$ and 8 users per class. The storage of each user is relatively small compared to the library of files. The coefficients of the system are close to the asymptotic regime in Section IV where $M < K << N_c = N_u$. The figures illustrate that Scheme 2, which splits the placement and delivery of common and unique files, achieves the lowest average rate among all three schemes. This result aligns with Proposition 3 and supports our hypothesis that Scheme 2 provides lower average rate than the other two for small M.

It is worth noting that these results are different from those observed in [10] for the peak rate, where Scheme 2 presented the highest peak rate among the three schemes for small M.

Figure 4 investigates the performance of the three schemes as the number of classes grows. In this scenario, $N_c = N_u = 64$, and we set $M = 64$ to provide enough storage for each user to cache half of the files it could request. The top plot stands for the case with 2 users per class, and the bottom plot for the case with 8 users per class. The figure shows that Scheme 2 achieves a lower average rate than the other two as G increases. This result aligns with the peak rate analysis in [10] and with Prop. 3. However, when G is small, the average rate with Scheme 2 is worse than that with the other two schemes, since the number of files does not fulfill the conditions of the aforementioned Proposition.

VI. Conclusion

The average rate is a critical metric for evaluating the performance of a coded caching system, since peak rate is scarcely reached during the system's operation. This paper derived the average rates for three caching schemes within a system with heterogeneous user profiles. Assuming that user requests are independent and uniformly distributed over their own demand set, it was shown that decoupling caching and delivery of common and unique files provides the lowest average rate than the other two schemes when the cache capacity is small. In addition, the comparison yields opposite trends for average rate and peak rate when the storage is small.

Moreover, we showed that when the size of the users' cache is small and the number of common files is larger than one-fourth of the total number of unique files, the users should invariably cache all the common files and transmit the unique files uncoded to achieve a minimal average rate. This scenario reflects practical systems where the storage capacity of users does not match up with the huge amount of online files. In addition, when one type of files significantly outnumbers the other, to reach the minimal average rate, the users are recommended to devote all of their cache storage to those files that are numerically larger.

Appendix

A. Proof of Lemma 1

Proof. We define the random variable I_j as the file demand indicator for each of the N files,

$$I_j = \begin{cases} 1 \text{ if at least one user requests } j\text{th file} \\ 0 \text{ otherwise.} \end{cases} \tag{17}$$

The total number of distinct files being requested by K users is then $X = \sum_{j=1}^{N} I_j$. Utilizing the linearity of expectation and assuming K independent requests uniformly distributed among the N files yields Eq. (11). □

B. Proof of Proposition 2

Proof. The partial derivative of $R_{\text{avg}}^{(2)}$ with respect to x is,

$$\frac{\partial R_{\text{avg}}^{(2)}}{\partial x} = \frac{\partial R_c}{\partial x} + \frac{\partial (GR_u)}{\partial x}. \tag{18}$$

In this equation, by using the chain rule to derive $R_{\text{avg}}^{(2)}$ with respect to t_c and t_u and then respect to x, the term which dominates the value of $\frac{\partial R_c}{\partial x}$ in terms of t_c is

$$Y_1 = \frac{\binom{K}{t_c+1}}{\binom{K}{t_c}} \frac{KM}{N_c} \Big[\psi(t_c+1) - \psi(t_c+2)\Big], \tag{19}$$

where $\psi(x)$ is digamma function and $t_c = \frac{KMx}{N_c}$. Similarly, the term that dominates the value of $\frac{\partial (GR_u)}{\partial x}$ in terms of t_u is

$$Y_2 = \frac{\binom{\frac{K}{G}}{t_u+1}}{\binom{\frac{K}{G}}{t_u}} \frac{KM}{N_u} \Big[\psi(t_u+2) - \psi(t_u+1)\Big], \tag{20}$$

where $t_u = \frac{KM(1-x)}{GN_u}$. Since $\psi'(x) > 0$, we can write that $Y_1 < 0, \forall x \in [0,1]$ and $Y_2 > 0, \forall x \in [0,1]$. $|Y_1|$ is minimal

when $x=1$, $|Y_2|$ reaches maximum when $x=1$. Hence, we need to prove that $|Y_1| \geq |Y_2|$ when $x=1$, so that $\frac{\partial R_{\text{avg}}^{(2)}}{\partial x} \leq 0, \forall x \in [0,1]$. The inequality can be written as

$$\frac{\binom{K}{\frac{KM}{N_c}+1}}{\binom{K}{\frac{KM}{N_c}}} \frac{KM}{N_c}\Big[\psi\Big(\frac{KM}{N_c}+2\Big)-\psi\Big(\frac{KM}{N_c}+1\Big)\Big] \geq \frac{K^2M}{GN_u}. \tag{21}$$

The left hand side is strictly increasing with $t_c = \frac{KM}{N_c}$. In this inequality, as long as $M \leq \frac{N_c}{K}$, we are able to let $\frac{KM}{N_c}=1$ which is the lowest t_c eligible. Hence, the inequality can be simplified to

$$M \leq \frac{GN_u - 2N_c}{2K + \frac{GN_u}{N_c}}. \tag{22}$$

Note that the assumption is that $M \leq \frac{N_c}{K}$, hence, the inequality $\frac{GN_u-2N_c}{2K+\frac{GN_u}{N_c}} \leq \frac{N_c}{K}$ must be satisfied. By simplifying the inequality, we are able to get the constraint which is $N_c \geq \frac{K-1}{4K} GN_u$.

□

C. Proof of Proposition 3

Proof. Under the asymptotic regime being considered ($N_c = N_u \to \infty$), it can be written that $N_c(\vec{d}) \to \frac{1}{2}K$, $N_u(\vec{d}) \to \frac{1}{2}\frac{K}{G}$. Therefore, by substituting both $N_c(\vec{d})$ and $N_u(\vec{d})$ in Eq. (12), expanding the combinatorial terms and taking the limit when $K \to \infty$, R_c is simplified as:

$$R_c = \frac{K - t_c - K(\frac{1}{2})^{t_c+1}}{t_c+1}. \tag{23}$$

We write $t_u = \frac{t_c}{G}\frac{1-x}{x}$, since $N_c = N_u$. Hence, R_u can be written as:

$$R_u = \frac{K - t_c\frac{1-x}{x} - K(\frac{1}{2})^{\frac{t_c}{G}\frac{1-x}{x}+1}}{\frac{t_c}{G}\frac{1-x}{x}+1}. \tag{24}$$

Hence, the average rate of Scheme 2 in asymptotic regime is:

$$R_{\text{avg}}^{(2)} = \frac{K - t_c - K(\frac{1}{2})^{t_c+1}}{t_c+1} + \frac{K - t_c\frac{1-x}{x} - K(\frac{1}{2})^{\frac{t_c}{G}\frac{1-x}{x}+1}}{\frac{t_c}{G}\frac{1-x}{x}+1}.$$

For Scheme 1 and 3, it can be written that $N_1(\vec{d}) \to K$ and $N_3(\vec{d}) \to K/G$. Therefore, by replacing $N_1(\vec{d})$, $N_3(\vec{d})$ and taking the limit while $K \to \infty$, the average rate for Scheme 1 and 3 can be written as:

$$R_{\text{avg}}^{(1)} = \frac{\binom{K}{t_1+1} - \binom{K-N_1(\vec{d})}{t_1+1}}{\binom{K}{t_1}} \to \frac{K-t_1}{t_1+1} \to K, \tag{25}$$

$$R_{\text{avg}}^{(3)} = G\frac{\binom{\frac{K}{G}}{t_3+1} - \binom{\frac{K}{G}-N_{3_i}(\vec{d})}{t_3+1}}{\binom{\frac{K}{G}}{t_3}} \to G\frac{\frac{K}{G}-t_3}{t_3+1} \to K. \tag{26}$$

$R_{\text{avg}}^{(2)}$'s minimum is reached when $x=1$, which was proved in the previous section. A simple calculation shows that $R_{\text{avg}}^{(2)} < K$ when $x=1$. □

D. Proof of Proposition 4

Proof. When $K << N_u << N_c \to \infty$, it can be expressed that $N_c(\vec{d}) \to K$, $N_u(\vec{d}) \to 0$. Therefore, by substituting $N_c(\vec{d})$ and $N_u(\vec{d})$ into Eq. (18), it can be calculated that

$$\frac{\partial R_{\text{avg}}^{(2)}}{\partial x} = \frac{\binom{K}{t_c+1}}{\binom{K}{t_c}}\frac{KM}{N_c}\Big[\psi(t_c+1)-\psi(t_c+2) +\psi(K-t_c)-\psi(K-t_c+1)\Big]. \tag{27}$$

Since $\psi'(x) > 0$, it can be written that $\frac{\partial R_{\text{avg}}^{(2)}}{\partial x} < 0, \forall x \in [0,1]$. Therefore, choosing $x=1$ will result in the minimal $R_{\text{avg}}^{(2)}$.

When $K << N_c << N_u \to \infty$, by replacing $N_c(\vec{d}) \to 0$ and $N_u(\vec{d}) \to \frac{K}{G}$ into Eq. (18), we get

$$\frac{\partial R_{\text{avg}}^{(2)}}{\partial x} = \frac{\binom{\frac{K}{G}}{t_u+1}}{\binom{\frac{K}{G}}{t_u}}\frac{KM}{N_u}\Big[\psi(t_u+2)-\psi(t_u+1) +\psi\Big(\frac{K}{G}-t_u+1\Big)-\psi\Big(\frac{K}{G}-t_u\Big)\Big]. \tag{28}$$

It can be derived that $\frac{\partial R_{\text{avg}}^{(2)}}{\partial x} > 0, \forall x \in [0,1]$. Hence, choosing $x=0$ will contribute to the minimal $R_{\text{avg}}^{(2)}$. □

REFERENCES

[1] M. A. Maddah-Ali and U. Niesen, "Fundamental limits of caching," IEEE Transactions on Information Theory, vol. 60, no. 5, pp. 2856–2867, 2014.

[2] U. Niesen and M. A. Maddah-Ali, "Coded caching with nonuniform demands," IEEE Transactions on Information Theory, vol. 63, no. 2, pp. 1146–1158, 2017.

[3] R. Pedarsani, M. A. Maddah-Ali, and U. Niesen, "Online coded caching," IEEE/ACM Transactions on Networking (TON), vol. 24, no. 2, pp. 836–845, 2016.

[4] T. Luo, V. Aggarwal, and B. Peleato, "Coded caching with distributed storage," arXiv preprint arXiv:1611.06591, 2016.

[5] Q. Yu, M. A. Maddah-Ali, and A. S. Avestimehr, "The exact rate-memory tradeoff for caching with uncoded prefetching," IEEE Transactions on Information Theory, vol. 64, no. 2, pp. 1281–1296, 2018.

[6] C.-H. Chang and C.-C. Wang, "Coded caching with heterogeneous file demand sets - the insufficiency of selfish coded caching," IEEE Internat. Symp. on Information Theory (ISIT), 2019.

[7] A. Hannak, P. Sapiezynski, A. Molavi Kakhki, B. Krishnamurthy, D. Lazer, A. Mislove, and C. Wilson, "Measuring personalization of web search," in Proceedings of the 22nd international conference on World Wide Web. ACM, 2013, pp. 527–538.

[8] A. McCallum, K. Nigam, J. Rennie, and K. Seymore, "A machine learning approach to building domain-specific search engines," in IJCAI, vol. 99. Citeseer, 1999, pp. 662–667.

[9] E. Agichtein, E. Brill, S. Dumais, and R. Ragno, "Learning user interaction models for predicting web search result preferences," in Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval. ACM, 2006, pp. 3–10.

[10] S. Wang and B. Peleato, "Coded caching with heterogeneous user profiles," IEEE Internat. Symp. on Information Theory (ISIT), 2019.

[11] C.-H. Chang and B. Peleato, "On coded caching for two users with overlapping demand sets," to appear in IEEE International Conf. on Comm. (ICC), 2020.

[12] C.-H. Chang, B. Peleato, and C.-C. Wang, "Coded caching with full heterogeneity: Exact capacity of the two-user/two-file case," submitted to IEEE Transactions on Information Theory.

[13] T. Luo and B. Peleato, "The transfer load-i/o trade-off for coded caching," IEEE Communications Letters, vol. 22, no. 8, pp. 1524–1527, 2018.