**Abstract**—Recently, a novel technique called coded cache have been proposed to facilitate the wireless content distribution. Assuming the users have the identical cache size, prior works have described the fundamental performance limits of such scheme. However, when heterogeneous user cache sizes is taken into account, there remains open questions regarding the performance of the cache. In this work, for the heterogenous user cache sizes, we first derive the information-theoretical lower bound and show that although the traditional coded caching scheme will cause miss of coding opportunities in this regime, it still exhibits the constant gap 6 to the lower bound. To reveal the essence in this result, we introduce the concept of the probabilistic cache set and find the guarantee of such order optimality mainly comes from the intrinsic degenerated efficiency of bottleneck caching network under the heterogenous cache sizes. Moreover, we point out such miss of coding opportunities is caused not only by the heterogenous cache sizes but also by the group coded delivery, which divide users into groups and operate coded cache separately. And the combination of above two schemes will not cumulate this effect, instead, they cancel each other out.

I. INTRODUCTION

The demand for high-definition video streaming such as YouTube and Netflix spurs the dramatic increase in mobile traffic volume over the wireless network. According to Cisco Visual Networking Index [1], the mobile wireless traffic volume beyond 2020 will be 1000 times higher than that in 2010. The conflict between the increasing demand for high quality media contents and limited wireless delivery capacities drives the communication society to develop new techniques rather than just reconstruct the infrastructure to tackle with the “Mobile data tsunami”.

The traffic in wireless network, especially in content delivery networks, shows strong temporal variability, where the network resources are extremely scarce in the peak hours and underutilized in the off-peak hours. It is therefore necessary to deploy the caching technique to balance the traffic load over the wireless link. In the idle period, the content server pushes the contents into the edge of wireless network. Then, in the peak time, if the user’s requested content can be found in a close-by cache memory, it can be served locally. The content server only sends the remaining parts using simple orthogonal unicast transmissions. The gain of this effort is dependent on the proportion of the total content can be cached locally and often referred as local cache gain. As the quantity and size of media content grows rapidly, this kind of gain will be increasingly meaningless.

Recently, Maddah-Ali [2] [3] proposes a novel caching technique, called coded cache, to further exploit the potential of the distributed cache resources and achieve a global cache gain, which can significantly reduce the network traffic rather than the uncode version. This technique not only delivers part of the content locally, but also creates the coding opportunities among different requests via jointly optimizing content placement and delivery phases. The setting considered in [2] [3] consists of a server connected to multiple users with the identical cache size. With this assumption, they derive the delivery-phase traffic volume and also theoretically prove that the required traffic volume over wireless only exhibits constant gap to the information-theoretical lower bound.

In practice, the cache sizes of different wireless equipments are heterogenous. For example, some caches is located in the wireless access point that can be assigned a $T_B$–level cache size, while others might be the user terminals that have a small cache size, even the different user terminals will have the different cache sizes due to the diversity of the modern mobile devices. The design and analysis of coded caching scheme under such practical scenario, i.e., heterogenous cache sizes, is still rare.

There are several key questions when designing and analyzing such coded caching scheme under heterogenous cache sizes. The first question is how to design the coded caching scheme in this regime? One particular point of interest is if the traditional coded caching schemes can be applied straightforwardly. In other words, if this scheme cannot be utilized, what’s the obstacle or whether there exists some methods to overcome this obstacle? The second question is how to characterize the optimal trade-off between the heterogenous cache sizes and the traffic volume produced by corresponding coded caching scheme. Can we get a simple form of such trade-off as same as [2] [3]? The last question is what is the information-theoretical lower bound of this regime and if above memory-traffic volume trade-off still exhibits constant gap.

In this work, we focus on a caching system consisted of $K$ users connecting to a server through a shared, error-free link. The server has a database of $N$ contents. Each user $i$ has accessed to a cache memory big enough to store $M_i$ of the contents. In the traditional coded caching scheme [3], each user has an identical memory space and the random caching procedure in the placement phase will produce the content segments of approximate equal size, and form a maximal clique of different segments that can be fully utilized to create the coding opportunities in the delivery phase. However, when the users’ cache sizes are heterogenous, it will produce the content segments of different sizes. One simplest approach is to pad the useless zero bits in the tail of each shorter segments to
coded caching also achieves an extra global cache gain of \( \sum_{i} \frac{\text{size of cache}}{\sum_{i} \text{size of cache}} \) which has a local cache gain that is dependent on aggregate size for coded caching based distribution and the rest in a distributed contents. We incorporate this scheme into the regime of heterogenous and homogenous cache size. The number of users \( K = 15 \), the number of contents \( N = 100 \). The aggregate cache size is identical under above two scenarios.

diminish such difference and adopt coded delivery scheme \[3\]. Apparently, this scheme will miss the coding opportunities for those longer segments and increase the delivery-phase traffic volume.

In this setting, the delivery-phase traffic volume of conventional uncoded caching scheme is

\[
\sum_{i=1}^{K} \left(1 - \frac{M_i}{N}\right) = K - \frac{\sum_{i=1}^{K} M_i}{N},
\]

which has a local cache gain that is dependent on aggregate size of cache \( \sum_{i=1}^{K} M_i \).

In contrast, our simple modified coded caching scheme proposed in this paper attains a traffic volume of

\[
\sum_{i=1}^{K} \left(1 - \frac{M_i}{N}\right) \prod_{j=1}^{i-1} \left(1 - \frac{M_j}{N}\right).
\]

Thus, in addition to the local cache gain of \( \sum_{i=1}^{K} M_i \), coded caching also achieves an extra global cache gain of \( \prod_{j=1}^{i-1} \left(1 - \frac{M_j}{N}\right) \).

Through deriving the fundamental bounds on the delivery-phase traffic volume in this regime, we show that: \textit{although this simply modified scheme will cause the “bits waste” phenomenon and miss the potential coding opportunities, it still exhibits a constant gap 6 to the optimal scheme for all } \( M, K \leq N \). The main reason is that the miss of coding opportunities is an inherent limitation of bottleneck network under the heterogenous cache sizes, which appears not only in the coded caching scheme, but also in the information-theoretical lower bound, as illustrated in Fig. 1.

Similarly, this kind of phenomenon also appears in the group coded delivery(GCD) scheme, which divides users into group and operate coded caching scheme separately. An intuition is if we incorporate this scheme into the regime of heterogenous cache sizes, whether such phenomenon will be cumulated. We show a surprising result that these two effect cannot cumulate together, instead, they actually cancel each other out, which means the heterogenous cache sizes will diminish the effect of GCD, the GCD will also diminish the effect of heterogenous cache sizes.

To further investigate the fundamental limits of heterogenous cache sizes, we introduce the concept of probabilistic cache set and characterize the memory-traffic volume tradeoff via the numerical statistics of the cache size distribution. We analytically show that both gaps of traffic volume produced by coded and uncoded caching scheme to the lower bound will decrease when the deviation of all users’ cache sizes increases, and the uncoded caching scheme shows a faster decreasing speed. This result implies: first, the coded caching scheme will gradually degenerate to the uncoded version as the difference among users’ cache sizes increases; second, the miss of coding opportunities is a intrinsic characteristic in the bottleneck caching network under heterogenous cache sizes, since the traffic volume of uncoded cache is irrelevant to deviation of cache sizes, the decreasing gap mainly comes from the increasing of information-theoretical bound.

The remainder of the paper is organized as follows. We describe the service model and problem setting in Section II and provide some preliminaries and motivations in Section III. Section IV presents our main results about coded cache under heterogenous cache sizes, including our modified coded caching scheme, traffic volume, group coded delivery and order optimality analysis. Section V further investigate this problem under the probabilistic cache set. Numerical analysis are presented in Section VI. Section VII concludes this work and exhibits some interesting extensions. The proof of our main results and details are provided in the Appendix.

II. MODEL AND ASSUMPTION

In this paper, we consider a set of users connecting to a content server through a shared wireless link that is similar as the setup in \[3\].

A. Problem Setting

We consider a network consists of a content server connected to \( K \) users through a shared, error-free link, as illustrated in Fig. 2. The error-free link can be achieved with error correction scheme or reliable transmission scheme in the upper layer. The user set is denoted by \( K = \{1, \ldots, K\} \). The content server has accessed to a database of \( N \) uniform distributed contents \( W_1, W_2, \ldots, W_N \) with each of size \( F \) bits.

The same-size-content assumption is for the theoretical convenience, which however does not hinder the practicability of the coded caching operations in the real world, because the main body of content objects can be tailored as the same size for coded caching based distribution and the rest in a small quantity can be distributed in the traditional way. The content index set is denoted by \( N = \{1, \ldots, N\} \). Each user \( k \) has an isolated cache space \( Z_k \) of size \( M_k F \) bits for some real number \( M_k \in [0, N] \). All users’ cache sizes constitutes a cache set \( M = \{M_1, \ldots, M_K\} \). Without loss of generality, we assume the cache set \( M \) is a ordered set,

\[1\]The size can also be packet-based. For simplicity, we use the bits as the metric in the following.

\[2\]Here we use the cache set to denote the set consisted of users’ cache sizes in stead of each user’s local cache \( Z_k \).
i.e., \( M_1 \leq M_2 \leq \ldots \leq M_K \). The system operates in two phases: a placement phase and a delivery phase.

**Definition 1:** (Achievable scheme)

\[
R_\mathcal{F} \triangleq \max_{(d_1, \ldots, d_K) \in \mathcal{N}^K} R_{\mathcal{F}}^{(d_1, \ldots, d_K)}
\]  

the worst case normalized traffic volume for scheme \( \mathcal{F} \).

We use the \( R^* \) to represent the smallest traffic volume such that \((\mathcal{M}, R^*)\) is achievable. Defined by

\[
\mathcal{M}, R^* = \inf \{(\mathcal{M}, R_\mathcal{F}), \forall \mathcal{M}, \mathcal{F}\}
\]  

the infimum of all achievable \((\mathcal{M}, R_\mathcal{F})\).

Clearly, \( R_\mathcal{F} \) is function of cache set \( \mathcal{M} \) and number of users \( K \) and number of contents \( N \). To emphasize this dependency, we rewrite above traffic volume as \( R_{\mathcal{F}}(\mathcal{M}, N, K) \) and \( R^*(\mathcal{M}, N, K) \). The aim of this paper is to find a scheme \( \mathcal{F} \) such that \( R_{\mathcal{F}}(\mathcal{M}, N, K) \) guarantees the order optimality.

**Definition 3:** (Order optimality) The scheme \( \mathcal{F} \) is order optimal if only if

\[
\frac{R_{\mathcal{F}}(\mathcal{M}, N, K)}{R^*(\mathcal{M}, N, K)} \leq C,
\]

\( C \) is a constant independent of these parameters.

We can see that the order optimality can be guaranteed only if the traffic volume produced by scheme \( \mathcal{F} \) has the constant gap to the optimum.

### III. Preliminary and Motivation

In this section, we review the basic coded caching scheme about more details and present the main motivation of our work.

**A. Related Works**

Coded caching is a novel technique to mitigate the wireless traffic volume during the peak-traffic time. This result has been extended to nonuniform demands in [4] and [5], online caching system in [6], hierarchical caching system in [7] and heterogeneous network with multi-level cache access in [8]. Since our work is mainly based on this seminal work [3], we now briefly review this scheme with an example so that the discussions in the following sections can be understood.

Remark that, we use the notation \( \tilde{\mathcal{F}} \) to represent this scheme.

**Example 1** (Decentralized Coded Caching) Suppose that there is a simple system distributing \( N = 2 \) contents \( A \) and \( B \) to \( K = 2 \) users, each with the cache size \( MF/2 \) bits. In the placement phase, each user randomly caches \( MF/2 \) bits of content \( A \) and \( B \) independently. Let us focus on content \( A \). The operations of placement phase partition content \( A \) into four subcontents, \( A = (A_0, A_1, A_2, A_{1,2}) \), where \( U \subset \{1, 2\} \), \( A_U \) denotes the segments of content \( A \) that are prefetched in the memories of users in \( U \). For example, \( A_1 \) represents the segments of \( A \) only available in the memory of user 1. We adopt \( | \cdot | \) as the operation of size, thus \( |A_0| = (1 - M/2)^2 F \) bits, \( |A_1| = |A_2| = (M/2)(1 - M/2)F \) bits and \( |A_{1,2}| = (M/2)^2 F \) bits. The same analysis holds for content \( B \).

In the delivery phase, we assume that user 1 and user 2 request content \( A \) and \( B \), respectively. User 1 has accessed subcontent \( A_1 \) and \( A_{1,2} \) in its local cache and lacks \( A_0 \) and \( A_2 \). Similarly, user 2 has already accessed \( B_2 \) and \( B_{1,2} \), and lacks \( B_0 \) and \( B_1 \). In traditional uncoded caching scheme, the

\[\text{Note that the size we discussed is the expected size of each segment.}\]
server is required to unicast $A_O$ and $A_2$ to user 1 and unicast $B_O$ and $B_1$ to user 2. The total traffic volume is

$$2 \left( \frac{M}{2} \right) \left( 1 - \frac{M}{2} \right) F + 2 \left( \frac{M}{2} \right)^2 F = (2 - M)F$$

With the coded caching scheme, the server can satisfy the requests by transmitting $A_O$, $B_O$ and $A_2 \oplus B_1$ over the shared link, where $\oplus$ denotes the bit-wise XOR operation. The traffic volume over the shared link is

$$\left( \frac{M}{2} \right) \left( 1 - \frac{M}{2} \right) F + 2 \left( \frac{M}{2} \right)^2 F = \left( 1 - \frac{M}{2} \right)^2 \left( 2 - M \right)F < (2 - M)F$$

We can see that, the coded caching scheme has a coded gain of $(1 - \frac{M}{2})$ compared to the uncoded caching scheme. The scheme for general case that has $N$ contents, $K$ users, each with cache size $M$ can be seen in Algorithm 1.

Algorithm 1: Decentralized Coded Caching Scheme $\mathcal{F}$.

**Placement Phase**

```
for (k = 0; k < K; k++) do
  for (n = 0; n < N; n++) do
    user k randomly prefetches $MF/N$ bits of content n;
```

**Delivery Phase**

```
for (k = 0; k < K; k++) do
  for choose k users from K users to form a user group $U$ do
    $X_U \leftarrow \oplus_{k \in U}V_k,k / (k)$;
    Multicast the coded data $X_U$ to users in $U$.
```

Operator $\oplus$ refers to bitwise XOR operation. In the placement phase, the random caching procedure will divide content $i$ into $2^K$ segments: $W_{i,k,U}, U \subset 2^K$. In the delivery phase, the element $V_{k,k,U}/(k)$ in signal $X_U$ represents the segment of user $k$’s requested content that being cached in users of set $U/k$.

The traffic volume produced by this scheme is

$$K \cdot \left( 1 - \frac{M}{N} \right) \cdot \frac{N}{KM} \cdot \left[ 1 - \left( 1 - \frac{M}{N} \right)^K \right]. \quad (6)$$

The term $K \cdot \left( 1 - \frac{M}{N} \right)$ can be seen as the traffic volume produced by uncoded caching scheme, and only contains the local cache gain that are linear to the cache size. While the coded caching scheme has another multiplicative term $\frac{N}{KM} \cdot \left[ 1 - \left( 1 - \frac{M}{N} \right)^K \right]$, named as the global cache gain, which is inverse proportional to the cache size.

**B. Motivation**

In the regime of heterogenous cache set, an intuition is whether the decentralized coded caching scheme $\mathcal{F}$ can be applied straightforward. In fact, making the following simple modifications of scheme $\mathcal{F}$, we can adopt it in our regime.

In the placement phase, user $k$ randomly prefetches $M_kF/N$ bits of content $n$.

Based on this caching strategy, the same content stored in different users’ local caches will occupy the different size of memory space. Thus, in the delivery phase, for each multicast user group $U$, the size of $V_k,k / (k)$ of each user $k$ will be different. Since these segments should participate bit-wise XOR, all the segments should be bits-padded to the longest one. A possible padding method is zero-bits-padding and we refer this scheme as $\mathcal{F}_{o}$. The following example illustrate the performance of scheme $\mathcal{F}_{o}$.

**Example 2** (Coded Caching with Heterogenous Cache Set)

Suppose that there are three similar system distributing $N$ contents to $K = 2$ users. The first system adopts our modified coded caching scheme $\mathcal{F}_{o}$ and the users have the cache set $\mathcal{M}_1 = \{(2 - \alpha)M, (2 + \alpha)M\}$ $F$ bits with $0 < \alpha < 2$. The second system divides users into two groups and adopts scheme $\mathcal{F}_{o}$ in each group. The users take the same cache set $\mathcal{M}_1$. The third system adopts traditional coded caching scheme $\mathcal{F}$, shown in Algorithm 1, and have the uniform cache size $\mathcal{M}_2 = \{2M, 2M\}F$ bits.

Remark that above three systems have the same aggregate cache size of $4MF$ bits. Our objective is to compare the traffic volume of above three systems.

Assume that user 1 and user 2 request content A and content B. Based on the coded delivery technique, the server transmitted signal $A_2 \oplus B_1, A_0$ and $B_0$, and the corresponding signal size via first system is

$$R_{\mathcal{F}_{o}}(\mathcal{M}_1, N, 2) = \max \left\{ \frac{(2 - \alpha)M}{N}, \frac{(2 + \alpha)M}{N} \right\} \left( 1 - \frac{(2 - \alpha)M}{N} \right) \left( 1 - \frac{(2 + \alpha)M}{N} \right) + 2 \left( 1 - \frac{(2 - \alpha)M}{N} \right) \left( 1 - \frac{(2 + \alpha)M}{N} \right) . \quad (7)$$

The corresponding signal size via second system is

$$R_{\mathcal{M}}(\mathcal{M}_1, N, 2) = \left( 1 - \frac{(2 - \alpha)M}{N} \right) + \left( 1 - \frac{(2 + \alpha)M}{N} \right) = \left( 2 - \frac{4M}{N} \right). \quad (8)$$

Based on the traffic volume formula, the corresponding signal size via third system is

$$R_{\mathcal{F}}(\mathcal{M}_2, N, 2) = \left( 1 - \frac{2M}{N} \right) \left( 2 - \frac{2M}{N} \right) . \quad (9)$$

Then we consider two kinds of scenarios: the cache set of small deviation ($\alpha = 0.2$) and the cache set of large deviation ($\alpha = 1.8$). The first scenario refers to the situation that the size of each user’s local cache is similar, i.e., $\mathcal{M}_1 = \{1.8M, 2.2M\}$. The second scenarios refers to another situation that the size of each user’s local cache is extremely different, i.e., $\mathcal{M}_1 = \{0.2M, 3.8M\}$. The number of contents $N = 4M$. 


Base on the traffic volume formula (7)-(9), we get the following results, shown in TABLE I.

| Scenario/Traffic(bits) | system 1 | system 2 | system 3 |
|------------------------|----------|----------|----------|
| α = 0.2                | 0.7975F  | 1F       | 0.75F    |
| α = 1.8                | 0.9975F  | 1F       | 0.75F    |

It can be seen that, when the size of each user’s local cache is similar, the traffic volume produced by system 1 approximates to the system 3 and much lower than system 2; when the size of each user’s local cache is large, it approximates to the system 2 and much larger than the system 3. This result comes from a “bits waste” phenomenon that when the user cache size varies widely, the size of requested segments A2, B1 are largely different, and the scheme $\mathcal{S}_o$ will pad lots of useless zero bits to the smaller segment B1, which will diminish part of coding opportunities for the longer segment A2 and thus increase the traffic volume.

Example 2 shows that when the cache set is approximately uniform, the coded caching scheme $\mathcal{S}_o$ still has a global cache gain, while when the cache set is skewing, this global gain will diminish and the grouping of users (system 2) will not reduce the performance of whole system. This phenomenon and the reason behind it inspires us in three dimensions. First, whether the scheme $\mathcal{S}_o$ is still order optimal or whether can we develop some new techniques to overcome it. Second, what condition the cache set should satisfy such that GCD will not increase the traffic volume compared to the non-grouping scheme; Third, whether there exists an efficient scheme can counteract this “bits waste” phenomenon and guarantee the order optimality independent of the characteristics of cache set. The following work answers these questions.

IV. CODED CACHING UNDER DETERMINISTIC CACHE SET

In this section, we first utilize the Fano’s inequality and cut-set-bound argument [16] to get the information-theoretical lower bound of the traffic volume, namely, the bound of $(M, R^*)$. Then we derive a close-form expression of traffic volume $R_{\mathcal{S}_o}(M, N, K)$ produced by our modified scheme $\mathcal{S}_o$, based on which we get a lot of interesting insights. Finally, we present the order optimality of scheme $\mathcal{S}_o$.

A. Information-theoretical lower bound

The information-theoretical lower bound is independent of any specific schemes, instead, only dependent on the system parameters including $M$, $N$ and $K$. The following theorem gives this lower bound on the optimal achievable traffic volume $R^*(M, N, K)$. It is a standard cut-set bound [16]: for a feasible $(M, R)$ pair, the total information contained in the memory of any subset of caches and the serve transmissions must be at least the size of contents that users accessing these caches can reconstruct.

**Theorem 1:** (Cut-set Bound) For caching problem with $N$ contents, $K (K \leq N)$ users, and ordered cache set $\mathcal{M} = \{M_1, M_1, \ldots, M_K\}$, we have

$$R^*(M, N, K) \geq R^*(M, N, K) \geq \max_{s \in \mathcal{K}} \left\{ \max_{U \subseteq [K], |U| = s} \left\{ s - \frac{\sum_{i \in U} M_i}{|N/s|} \right\} \right\}.$$  \hspace{1cm} (10)

**Proof:** Let $s \in \{1, \ldots, \min\{N, K\}\}$ and consider cache set consisted of $s$ user cache: $Z_u = \{Z_i, i \in U\} | U \subseteq K, |U| = s$. Note that cache set $Z_u$ is one of $C_K$ subsets when there are $s$ caches. Assume there are $s$ users send their requests and a corresponding transmitted signal in the shared link, termed as $X_1$, such that $X_1$ and $Z_u$ determine the contents $W_1, \ldots, W_s$. Similarly, there is a transmitted signal in the shared link, termed as $X_2$, such that $X_2$ and $Z_u$ determine the contents $W_{s+1}, \ldots, W_{2s}$. Under the same manner, we can use $X_1, \ldots, X_{N/s}$ and $Z_u$ to determine $W_1, \ldots, W_{s[N/s]}$.

Consider $\lfloor N/s \rfloor$ broadcasting signals $X_1, \ldots, X_{N/s}$ that satisfying the requests of $s$ users for $s|N/s|$ contents. Then, by Fano’s inequality

$$\frac{N}{s} R^*(M, N, K) + \sum_{i \in U} M_i \geq H \left( Z_u, X_1, \ldots, \frac{X}{r} \right) \geq H \left( Z_u, X_1, \ldots, \frac{X}{r} \right) \left| W_1, \ldots, W_{s[N/s]} \right| + s \left\lfloor \frac{N}{s} \right\rfloor \left( 1 - \varepsilon_p \right) \geq s \left\lfloor \frac{N}{s} \right\rfloor \left( 1 - \varepsilon_p \right).$$ \hspace{1cm} (11)

By taking $F \rightarrow \infty, \varepsilon_p \rightarrow 0$, and Solving $R^*(M, N, K)$ and optimizing over all possible choices of $s$, we can get

$$R^*(M, N, K) \geq \max_{s \in \mathcal{K}} \left\{ \max_{U \subseteq [K], |U| = s} \left\{ s - \frac{\sum_{i \in U} M_i}{|N/s|} \right\} \right\},$$  \hspace{1cm} (12)

proving the theorem.

In fact, previous work [2] point out the cut-set bound is sometimes loose, and tighter bounds on $R^*(M, N, K)$ can be derived via stronger arguments than the cut set bound. There are some works investigating how to improve this lower bound [2] [9]. For example, in the Appendix of [2], they show a possible method to get a sharper bound for $K = 2, N = 2$. Using this method, we can get the following bound in our regime

$$R^*(M, 2, 2) \geq \max \left\{ \frac{3}{2} - \frac{M_1 + M_2}{2}, 1 - \frac{M_1}{2} \right\}.$$  \hspace{1cm} (12)

that is much sharper than cut set bound [12]. However, the cut-set bound along is sufficient for illustration of the main idea of our work and we will use this bound in the following discussion.

We refer $\mathcal{M}^l$ as the homogeneous cache set for the heterogeneous cache set $\mathcal{M}$, both of which have the identical aggregate cache size. Defined by

**Definition 4:** (Homogeneous cache set) The average set of $\mathcal{M}$ is defined as $\mathcal{M}^l = \{\mathcal{M}, \mathcal{M}, \ldots, \mathcal{M}\}$, where $\mathcal{M} = \mathbb{E}_k[M_k]$. 

**TABLE I**

**THE CODED DATA AND TRAFFIC VOLUME INCURRED**

| Scenario/Traffic(bits) | system 1 | system 2 | system 3 |
|------------------------|----------|----------|----------|
| α = 0.2                | 0.7975F  | 1F       | 0.75F    |
| α = 1.8                | 0.9975F  | 1F       | 0.75F    |
Then we have the following relationship between two kinds of lower bounds.

**Corollary 1:** (Relation between two kinds of cut-set bound) For caching problem with $N$ contents, $K(K \leq N)$ users with ordered cache set $\mathcal{M} = \{M_1, M_2, \ldots, M_K\}$, we have

$$R^c(\mathcal{M}, N, K) \geq R^c(\mathcal{M}', N, K), \quad \text{(13)}$$

equal if only if $\mathcal{M} = \mathcal{M}'$.

As can be seen in Corollary 1 when the aggregate cache size is fixed, the cut-set bound under heterogenous cache set is larger than the cut-set bound under homogenous cache set. Namely, the cut-set bound will increases when the aggregate cache sizes is distributed in a nonuniform manner. This result can be regarded as a preliminary illustration for the degenerated performance under heterogenous cache set. The reason will be discussed in the last subsection of Section IV and the further quantitative discussion will be presented in the Section VIII. Complete proof is in Appendix VIII-A.

**B. Traffic volume of Zero-Bit-Padding Scheme $\mathcal{F}_o$**

For clarity, we rewrite the main procedure of our modified scheme $\mathcal{F}_o$ mentioned before. The pseudocode is shown in Algorithm 2.

**Algorithm 2:** Decentralized coded caching scheme with nonuniform cache size $\mathcal{M}$.

**Placement Phase**

for $(k = 0; k < K; k++)$

for $(n = 0; n < N; n++)$

user $k$ randomly prefetches $M_k F/N$ bits of content $n$;

**Delivery Phase**

for $(k = K; k > 0; k--)$

choose $k$ users from $K$ users to form a subset $U$;

for $l \in U$

if $|V_{k,U/(l)}| < \text{Maxsize}$ then

$\text{temp} \leftarrow (\text{Maxsize} - |V_{k,U/(l)}|)$ bits of all zero;

$V_{k,U/(l)} \leftarrow V_{k,U/(l)} + \text{temp}$;

$X_U \leftarrow X_U \oplus V_{k,U/(l)}$;

Multicast the coded data $X_U$ to users in $U$.

$\text{Maxsize} \leftarrow \max_{k \in U} |V_{k,U/\{k\}}|$;

$\text{Proof:}$ Consider $p \in U$ and $p \geq q = \inf U$, if we prove $|V_{q,U/(q)}| \geq |V_{p,U/(p)}|$, we can prove this Lemma. Consider a particular bit of $V_{q,U/(q)}$, the probability for this bit to be stored by the users of $U/(q)$ and not stored by the others is

$$\prod_{i \in U/(q)} \left( \frac{M_i}{N} \right) \prod_{i \in \{K/\} U} \left( 1 - \frac{M_i}{N} \right) F \quad \text{(15)}$$

Thus, the size of $V_{q,U/(q)}$ is

$$|V_{q,U/(q)}|F = \prod_{i \in U/(p,q)} \left( \frac{M_i}{N} \right) \prod_{i \in \{K/\} U} \left( 1 - \frac{M_i}{N} \right) \left( 1 - \frac{M_p}{N} \right) \left( \frac{M_q}{N} \right) F.$$

Since $p \geq q$ and $M_p \geq M_q$ (the ordered property of cache set $\mathcal{M}$), we have

$$\left( 1 - \frac{M_q}{N} \right) \left( \frac{M_p}{N} \right) \geq \left( 1 - \frac{M_p}{N} \right) \left( \frac{M_q}{N} \right).$$

Hence,

$$|V_{q,U/(q)}|F \geq \prod_{i \in U/(p,q)} \left( \frac{M_i}{N} \right) \prod_{i \in \{K/\} U} \left( 1 - \frac{M_i}{N} \right) \left( 1 - \frac{M_p}{N} \right) \left( \frac{M_q}{N} \right) F.$$

**Lemma 2:** (Traffic formula) For caching problem with $N$ contents, $K(K \leq N)$ users each with ordered cache set $\mathcal{M} = \{M_1, M_2, \ldots, M_K\}$, the Algorithm 1 produced the traffic,

$$R_{\mathcal{F}_o}(\mathcal{M}, N, K) = \sum_{s=1}^{K} \sum_{i=1}^{s} P_i Q_i \left( \bigotimes_{j=s-i+1}^{N} S_i \right). \quad \text{(19)}$$

where

$$P_i = \prod_{j=i+1}^{K} \frac{M_j}{N}, \quad 1 \leq i < K; i = K; \quad \text{(20)}$$

$$Q_i = \prod_{j=1}^{i} \left( 1 - \frac{M_i}{N} \right), \quad 1 \leq i \leq K; \quad \text{(21)}$$

$$S_i = \sum_{j=i+1}^{K} \left( \frac{N - M_j}{M_j} \right), \quad 1 \leq i \leq K; \quad \text{(22)}$$

Operator $+$ refers to bits concatenation.

Since we take the zero-bits-padding for each segment before delivery, the size of signal $X_U$ is determined by the longest segment $V_{k,U/(k)}$, $k \in U$. The following Lemma shows an important characteristic of this longest segment.

**Lemma 1:** (Ordered segment size) Based on Algorithm 2, the size of each transmission signal $X_U$ is,

$$|X_U| = \max_{k \in U} |V_{k,U/(k)}| = |V_{\inf \cup U/(\inf U)}|. \quad \text{(14)}$$
\[ R_{\mathcal{S}_o}(\mathcal{M}^{(K)}, N, K) = T_1^{(K)} + \sum_{s=2}^{K-1} T_s^{(K)} + (K + 1) \prod_{i=1}^{K} \left( 1 - \frac{M_i}{N} \right) \]

\[ = \frac{M_K}{N} T_1^{(K-1)} + \sum_{s=2}^{K-1} \left[ \frac{M_K}{N} T_s^{(K-1)} + \left( 1 - \frac{M_K}{N} \right) T_{s-1}^{(K-1)} \right] + (K + 1) \prod_{i=1}^{K} \left( 1 - \frac{M_i}{N} \right) \]

\[ = \frac{M_K}{N} R(\mathcal{M}^{(K-1)}, N, K-1) + \left( 1 - \frac{M_K}{N} \right) \left[ R(\mathcal{M}^{(K-1)}, N, K-1) - T_{K-1}^{(K-1)} \right] + (K + 1) \prod_{i=1}^{K} \left( 1 - \frac{M_i}{N} \right) \]

\[ = R(\mathcal{M}^{(K-1)}, N, K-1) + \prod_{i=1}^{K} \left( 1 - \frac{M_i}{N} \right) \]

(16)

The operation (b) is based on the assumption when \( k = K - 1 \), seen in (25).

Lemma 2 presents a possible formula to calculate the traffic volume under ordered cache set \( \mathcal{M} \). The complete proof can be seen in Appendix [VII-B]. We use the following toy example to illustrate the elite of Lemma 1 and Lemma 2.

**Example 3** (\( K = 3, \mathcal{M} = \{ M_1, M_2, M_3 \} \)) Consider the caching problem with \( N \) contents and \( K = 3 \) users with cache set \( \{ M_1, M_2, M_3 \} \). Assume the users request content A, B and C in the delivery phase. The transmission signal and corresponding signal size are,

- \( |W_{1,23} \oplus W_{2,13} \oplus W_{3,12}| = (1 - \frac{M_1}{N}) \frac{M_2}{N} \frac{M_3}{N} \); 
- \( |W_{1,2} \oplus W_{2,1}| = (1 - \frac{M_2}{N}) \frac{M_1}{N} \frac{M_3}{N} \); 
- \( |W_{3,1} \oplus W_{1,3}| = (1 - \frac{M_3}{N}) \frac{M_1}{N} \frac{M_2}{N} \); 
- \( |W_{2,3} \oplus W_{3,2}| = (1 - \frac{M_2}{N}) \frac{M_1}{N} \frac{M_3}{N} \); 
- \( |W_{1,0} \oplus W_{2,0} \oplus W_{3,0}| = 3 \left( 1 - \frac{M_1}{N} \right) \left( 1 - \frac{M_2}{N} \right) \left( 1 - \frac{M_3}{N} \right) \).

The operation (a) is based on the Lemma 1. After simplification, the total traffic volume is 

\[ P_1 Q_1 + \]

\[ P_1 Q_1 S_1 + P_2 Q_2 + \]

\[ P_1 Q_1 (S_1 \otimes S_2) + P_2 Q_2 S_1 + P_3 Q_3. \]

In fact, the traffic formula in Lemma 2 is difficult to calculate due to the combinatorial term (23), i.e., it has \( \binom{K}{s-i} \) terms. However, this formula shows an useful iterative structure that we can utilize to get a close-form expression of the traffic volume under scheme \( \mathcal{S}_o \).

**Theorem 2**: (Close-form expression of the traffic formula) For caching problem with \( N \) contents, \( K (K < N) \) users, with ordered cache set \( \mathcal{M} = \{ M_1, M_1, \ldots, M_K \} \), the scheme \( \mathcal{S}_o \) produces the traffic volume of,

\[ R_{\mathcal{S}_o}(\mathcal{M}, N, K) = \prod_{i=1}^{K} \left( 1 - \frac{M_i}{N} \right) \]

(24)

**Proof**: We use the mathematical induction to prove the following proposition.

\[ P(k) : \] A caching problem with \( N \) contents, \( k \) users, when the cache set \( \mathcal{M}^{(k)} \) satisfies \( M_1 < M_2 < \cdots < M_k \), the traffic volume produced by scheme \( \mathcal{S}_o \) is (24).

For clarity, we make the induction from \( k = 2 \) and \( P(1) \) is apparently correct.

1) Base step: \( k = 2 \), based on Example 2, the traffic volume is

\[ R_{\mathcal{S}_o}(\mathcal{M}^{(2)}, N, 2) = \left( 1 - \frac{M_1}{N} \right) \left( 1 - \frac{M_2}{N} \right) \]

(17)

2) Induction step: Assume, \( k = K - 1 \), the traffic volume is

\[ R_{\mathcal{S}_o}(\mathcal{M}^{(K-1)}, N, K - 1) = \prod_{i=1}^{K-1} \left( 1 - \frac{M_i}{N} \right) \]

(25)

Then we calculate the traffic volume when \( k = K \). Based on Lemma 2,

\[ R_{\mathcal{S}_o}(\mathcal{M}^{(K)}, N, K) = \prod_{s=1}^{K} T_s^{(K)}, \]

where \( T_s^{(K)} = \sum_{i=1}^{s} P_i^{(K)} Q_s^{(K)} \left( \otimes S_{s+1}^{(K)} \right) \). Here we use the superscript \( (K) \) to denote the operator of (16)-(22) when there are \( K \) users. The induction is proved by (16)-(25).

The operation (a) is based on the structure of operator \( P_i, Q_s, S_i \) and \( \otimes S_i \). Denote by following two situations.

When \( s = 1 \),

\[ T_1^{(K)} = P_1^{(K)} Q_1^{(K)} = \frac{M_K}{N} P_1^{(K-1)} Q_1^{(K-1)} \]
When $1 < s < K$, 
\[ T_s^{(K)} = \sum_{i=1}^{s} P_i^{(K)} Q_i^{(K)} \left( \bigotimes_{s-i+1}^{K} S_i^{(K)} \right) \]
\[ = \frac{M_K}{N} \sum_{i=1}^{s} P_{i-1}^{(K)} Q_{i-1}^{(K)} \left( \bigotimes_{s-i+1}^{K} S_i^{(K)} \right) \]
\[ = \frac{M_K}{N} \sum_{i=1}^{s} P_{i-1}^{(K)} Q_{i-1}^{(K)} \left( \bigotimes_{s-i+1}^{K} S_i^{(K)} \right) \]
\[ + \frac{M_K}{N} \cdot \frac{N - M_K}{M_K} \sum_{i=1}^{s-1} P_{i-1}^{(K)} Q_{i-1}^{(K)} \left( \bigotimes_{s-i}^{K} S_i^{(K)} \right) \]
\[ = \frac{M_K}{N} T_s^{(K-1)} + \left( 1 - \frac{M_K}{N} \right) T_s^{(K-1)} . \]

3) Combining base step and induction step, we conclude by the Principle of Mathematical Induction that \( F(k) \) is true for each positive \( k \) and ordered cache set \( M(k) \). Let \( k = K \), we prove this Theorem.

Both Lemma 2 and Theorem 2 present the formula to calculate the traffic volume of scheme \( S_o \). Lemma 2 calculate the traffic volume based on the transmissions, which seems like kind of a “top–bottom” portrait, while the Theorem 2 calculate the traffic volume based on the users, which seems like kind of a “left–right” portrait.

Besides, the proof of Theorem 2 and the traffic volume formula (24) provide us with some observations: in the aspects of sth transmission, the traffic volume due to the introduction of user \( K \) can be seen as the average combination of \( (s-1) \)th and sth transmissions when there are \( (K-1) \) users; in the aspects of the total traffic formula, the increment is denoted by following term
\[ \prod_{i=1}^{K} \left( 1 - \frac{M_i}{N} \right) . \tag{26} \]

The main reason is that the random caching in the placement phase and coded multicasting procedure in the delivery phase “connects” the users’ local caches together, such that the requested sent from a new user can be not only satisfied by its own cache but also by other users’ caches. We use the following equivalent network model, caching and delivery scheme to present another interesting illustration.

As shown in the right part of Fig. 3, the equivalent network diagram consists of a content server connecting to \( K \) users through an error-free unicasting link. Each user can utilize the contents that being stored in the users having smaller cache size, and communicate with them by an unicasting link. The caching procedure is same as scheme \( S_o \) and the delivery procedure is listed in the Algorithm 3.

Algorithm 3: Equivalent scheme \( S_u \) of scheme \( S_o \).

**Delivery Phase**

for \( (k = K, k > 0; k = -) \) do

for \( (i = k, i > 0; i = -) \) do

\[ \text{Sever sends the rest part of } W_{d_k} \text{ to user } k; \]

\[ \text{Sever sends the rest part of } W_{d_k} \text{ to user } k; \]

The operation \( W_{d_k} \cap Z_i \) in Algorithm 3 represents the part of content \( W_{d_k} \) that stored in user \( i \)'s local cache. The delivery phase of scheme \( S_u \) uses the hierarchical unicasting transmission. When user \( k \) sends its request \( d_k \), it first uses its local cache to reconstruct part of its requested content \( W_{d_k} \), then it successively uses user \( (k-1) \) to \( 1 \)'s local cache to reconstruct \( W_{d_k} \), finally the server unicasts the part of content \( W_{d_k} \) that are not stored in user \( 1 \) to \( k \)'s local cache to user \( k \). Thus, the traffic volume incurred in the unicasting link between server and user \( k \) is \( 26 \). Summing the traffic volume produced by all users, we can get the traffic formula that is identical to (24).

From this equivalent network diagram and scheme \( S_u \), we can see more clearly how the coded multicast plays a role in “connecting” user’s local cache. For the traditional uncoded cache, each user only uses its local cache, while for the scheme \( S_o \), each user can access other user’s cache that has a smaller cache size.

We then use the simple traffic volume formula above to derive some meaningful results.

**Corollary 2:** (Traffic volume under uniform cache set)
When the size of each user’s local cache all equals to \( M \), and the corresponding cache set is \( M_u = \{ M, \ldots, M \} \), the number of users is \( K(K < N) \), we have,
\[ R_{S_o}(M_u, N, K) = K \left( 1 - \frac{M}{N} \right) \frac{N}{KM} \left[ 1 - \left( 1 - \frac{M}{N} \right)^{K} \right] , \]
and
\[ R_{S_o}(M_u, N, K) \leq 12 . \]

It can be seen that, the traffic formula is same as (6), which has the same order of the information theoretical lower bound. The proof is in Appendix VIII-C.

**Corollary 3:** (Traffic volume under singularity cache size)
When the the cache set is \( M_s = \{ M, M, \ldots, (1+\alpha)M \} \), \( \alpha > 0 \), and the number of users is \( K(K < N) \), we have,
\[ R_{S_o}(M_s, N, K) = R(M_u, N, K) - \alpha \frac{M}{N} \left( 1 - \frac{M}{N} \right)^{K-1} . \tag{27} \]
Corollary 2 shows that, the increase of aggregate cache size $KM$ will reduce the traffic volume in an inverse proportional manner, as show in the term $N/KM$. However, Corollary 3 shows that the increase of aggregate cache size $(K + \alpha)M$, i.e., the increase of parameter $\alpha$, will reduce the traffic in a linear manner. Besides, another observation is, although there exists the “bit waste” phenomenon, the increase of a single cache size will also decrease the traffic volume. The proof can be seen in Appendix VIII-D.

**Corollary 4:** (Traffic volume under two groups of cache size) When the number of users is $K(K < N)$, the cache set is $\mathcal{M}_d = \{M_1, M_2\}$, where $M_1 = \{\alpha M, \ldots, M\}$, $M_2 = \{\alpha M \ldots \alpha M\}$, $\alpha > 0$, and $|M_1| = |M_2| = K/2$ (Assume $K$ is an even number), we have,

$$R_{\delta_n}(M_d, N, K) = R_{\delta_n}(M_1, N, K/2) + Q_{\delta_n}^n R(M_2, N, K/2),$$

where $Q_{\delta_n}^n = \left(1 - \frac{M}{N}\right)^{K/2}$.

In Example 2, when the cache set has large deviation, the GCD only increases a little traffic volume. Here Corollary 4 makes a analytical explanation for this result. Since $\alpha M \leq N$, we have $Q_{\delta_n}^n \geq \left(1 - \frac{1}{\alpha}\right)^{K/2}$. Then the cache set $\mathcal{M}_d$ has a large deviation corresponds to the large value of parameter $\alpha$ and that $Q_{\delta_n}^n$ approximates to 1. Thus, the traffic formula $R_{\delta_n}(M_d, N, K)$ will approximate to the traffic produced by two groups. The proof can be seen in Appendix VIII-E.

The following corollary shows the relationship between traffic volume under heterogenous cache set and homogenous cache set, which is same as Corollary 4.

**Corollary 5:** (Traffic volume uniformitarian argument) For caching problem with $N$ contents, $K(K \leq N)$ users with ordered cache set $\mathcal{M} = \{M_1, M_2, \ldots, M_K\}$, we have,

$$R_{\delta_n}(\mathcal{M}, N, K) \geq R_{\delta_n}(\mathcal{M}^1, N, K),$$

equal if only if $M_k = E_k[M_k], \forall k \in K$.

Corollary 5 states a fact that, when the system aggregation cache size is fixed, the uniform distribution of user cache size will accomplish the minimal traffic volume under the scheme $\mathcal{M}$. This is easy to understand for the scheme $\mathcal{M}$, since there is no “bit waste” phenomenon when the size of each user’s local cache is identical, the traffic volume will be minimized. The proof can be seen in Appendix VIII-F.

**C. Group Coded Delivery under Heterogenous Cache Set**

In the delay-sensitive regime, such as video-on-demand streaming, each user will have the delay constraint and cannot wait for the arrival of all users’ requests. Thus, we should divide the users into groups based on their delay tolerant threshold, i.e., users with high threshold are in the large group, while the users with low threshold are in the small group. Then operate the scheme $\mathcal{M}_d$ in each group separately. This strategy will give rise to the miss of coding opportunities among the requests of different groups. We use the following toy example to illustrate.

**Example 4** (Group coded delivery) Consider two simple caching systems, the first is same as Example 3, the second divide users into two groups $\{M_1, M_2\}$ and $\{M_3\}$. Under the first system, the content $A$, $B$, and $C$ are divided into 8 segments, separately. Let us focus on content $A$.

$$A = \{A_1, A_2, A_3, A_{12}, A_{13}, A_{23}, A_{123}, A_o\}.~(*)$$

while under the second system, each content is divided into 4 segments in the first group and 2 segments in the second group.

$$A' = \{A_{1}', A_{2}', A_{12}', A_o'\}, \quad A'' = \{A_{1}''', A_o''\}.~(**)$$

Consider the meaning of above logical segments, we have the following one-to-multiple correspondence.

- $A_1' = \{A_1, A_{13}\}$, $A_2' = \{A_2, A_{23}\}$;
- $A_{12}' = \{A_{12}, A_{123}\}$, $A_o' = \{A_3, A_o\}$;
- $A_3' = \{A_{13}, A_{23}, A_{123}\}$, $A_o'' = \{A_1, A_2, A_{12}, A_o\}$.

Then assume the users request content $A$, $B$, and $C$, separately. The first system will send the following signals

$$A_2 \oplus B_{13} \oplus C_{12}, A_2 \oplus B_1, A_3 \oplus C_1, B_3 \oplus C_2, A_o, B_o, C_o.$$  

The second system will send

$$A_2' \oplus B_1', A_1', A_o', C_o' \triangleq \{A_2, A_{23}\} \oplus \{B_1, B_{13}\}$$

$$\{A_3, A_o\}, \{B_3, B_o\}, \{C_1, C_2, C_{12}, C_o\}.$$  

Comparing these two sequences of signals, we can find that the segments of the boldface in the second system miss the coding opportunities. Thus the traffic volume in the second system will be larger than the first system. However, when we consider the miss of coding opportunities due to the heterogenous cache set, we can find the zero-bit-padding is appeared in the segments $B_{13}, C_{12}, B_1, C_1, C_2$ of first system, while just appeared in the segments $B_1, B_{13}$ of second system, which implies that GCD can diminish the effect of the heterogenous cache set.

To further investigate this phenomenon, we first define the miss coding ratio of heterogenous cache set and GCD, which can be regarded as the ratio of the traffic volume between them and original system.

**Definition 5:** (Miss coding ratio of heterogenous cache set) The miss of coding opportunities due to the heterogenous cache set $\mathcal{M}$ is defined as

$$M_H(\mathcal{M}, N, K) = \frac{R_{\delta_n}(\mathcal{M}, N, K)}{R_{\delta_n}(\mathcal{M}^1, N, K)}.\quad (29)$$

**Definition 6:** (Miss coding ratio of GCD) The miss of coding opportunities due to the heterogenous cache set $\mathcal{M}$ is defined as

$$M_G(\mathcal{M}, N, K, L) = \sum_{i=1}^{L} \frac{R_{\delta_n}(M_i, N, \lfloor K_i \rfloor)}{R_{\delta_n}(\mathcal{M}, N, \lfloor K \rfloor)}.\quad (30)$$

Based on above definition, the strict illustration that the GCD and the heterogenous cache set will cancel each other out will be transformed into the analytical proof of following: (1) the miss coding ratio of GCD under the heterogenous cache set is less than that under the homogenous cache set; (2) the
miss coding ratio of heterogeneous cache set under the GCD will be less than that without the GCD.

**Theorem 3**: (GCD diminishes the effect of heterogeneous cache set) For caching problem with $N$ contents, $K (K < N)$ users with ordered cache set $\mathcal{M} = \{M_1, M_1, \ldots, M_K\}$, we divide all users into $L (L > 1)$ groups $\mathcal{K} = \{K_1, \ldots, K_L\}$, and corresponding partition of cache set is $\mathcal{M} = \{M_1, \ldots, M_L\}$, then we have,

$$\frac{1}{K} \sum_{i=1}^{L} |K_i| \cdot m_H(M_i, N, |K_i|) \leq m_H(\mathcal{M}, N, K). \quad (31)$$

The equal condition of the first inequality is that if only if $M_i = \mathcal{M}, \forall i$.

The right side in **Theorem 3** can be regarded as the average miss coding ratio of the heterogeneous cache set. The complete proof can be seen in Appendix.

**Theorem 4**: (Heterogeneous diminishes the effect of GCD) For caching problem with $N$ contents, $K (K < N)$ users with ordered cache set $\mathcal{M} = \{M_1, M_1, \ldots, M_K\}$, we divide all users into $L$ groups $\mathcal{K} = \{K_1, \ldots, K_L\}$, and corresponding partition of cache set is $\mathcal{M} = \{M_1, \ldots, M_L\}$, then we have,

$$1 \leq m_G(\mathcal{M}, N, K) \leq m_G(M^*, N, K, L) < L. \quad (32)$$

The equal condition of the first inequality is that if only if $M_i = 0, \forall i \in \mathcal{M}^0$, where $\mathcal{M}^0$ is the zero size cache set, defined as, $\mathcal{M}^0 = \{M_i | 1 \leq i \leq \max_{K \notin \mathcal{K}} \{	ext{sup} K_i\}\}$. The equal condition of the second inequality is that if only if $M_i = \mathcal{M}, \forall i$.

The first inequality in **Theorem 4** shows that the GCD will increase the total traffic volume compared with the original system. The equal condition answers our second question in Section III that only if the cache size of user 1 to user $\max_{K \notin \mathcal{K}} \{	ext{sup} K_i\}$ is zero, the grouping of users will not increase the traffic volume. Let us relax this condition and show some enlightenment from it. If the grouping of users is based on the size of their cache, i.e., classify the users that having small cache size into the first $L - 1$ groups and those having large cache size into the last group, in this case, only if the cache size of users in the first $L - 1$ groups is extremely less than the users in the last group, the grouping method will increase little traffic volume. The $m_G(\mathcal{M}^*, N, K, L)$ can be seen as the upper bounds of GCD. The complete proof can be seen in Appendix VII-G.

Moreover, **Theorem 4** provides us with an insight that the grouping method under heterogeneous cache set will produce extra performance gain compared with the homogenous cache set, and a loose bound $L$ is presented. If $L = \Theta (K)$, we can get a linear complexity $O(K)$ coded caching scheme that cannot guarantee the order optimality since it will introduce at most $L$ times the traffic volume. In fact, if we can get a tight bound such that independent of the system parameters $N, K$ and $L$, we can straightforwardly get a linear complexity coded caching scheme that can guarantee the order optimality. This is further discussed in next two sections.

**D. Order Optimality**

**Theorem 5**: (Order optimality of scheme $\mathcal{F}_o$) For caching problem with $N$ contents, $K (K \leq N)$ users with heterogeneous cache set $\mathcal{M} = \{M_1, M_1, \ldots, M_K\}$, we have,

$$\frac{R_{\mathcal{F}_o}(M, N, K)}{R^*(\mathcal{M}, N, K)} \leq 6. \quad (33)$$

It can be seen that the scheme $\mathcal{F}_o$ exhibits the constant gap 6 to the cut-set bound, even less than the uniform case. As analyzed before, since the miss of coding opportunities in this regime will lead to the increase of the traffic volume, the only reason for the decreasing constant gap is that the increase of cut-set bound, i.e., a more increasing speed due to the heterogenous cache set. Here we discuss the main reason behind this trend. The cut-set bound argument presents a essence that the broadcasting signal of caching network plays a role in combining users’ caches to joint reconstruct their requested contents, and for a user group $U$ of $s$ users, the minimum size of broadcasting signal will always be bounded by those $s$ minimum caches. If the cache sizes are distributed in a nonuniform manner, the gain coming from such combining information retrieval will be weaken, namely, the broadcasting signal is limited in this regime. Thus, although coding scheme $\mathcal{F}_o$ will miss the coding opportunities, it still guarantee the order optimality.

Moreover, this constant gap is related to the deviation of the cache set. Consider two extreme cases: in the first case, the cache set has the smallest deviation that the all users’ cache size is identical, then the gap has been proved to 4; in the second case, the cache set has the largest deviation that the cache size of half users is 0 while another half is $N$, then the traffic volume and the corresponding information-theoretical lower bound is all $K/2$, the constant gap reduced to 1. In the next Section, we will use the concept of probabilistic cache set to quantitatively investigate the relationship between this constant gap and the characteristics of the cache set.

**V. CODED CACHING UNDER PROBABILISTIC CACHE SET**

In this section, we assume the cache size of each user is a random variable and the corresponding cache set is referred to probabilistic cache set. We first rewrite the basic definitions of this problem under probabilistic cache set, including the definition of expected optimal memory-traffic volume pair and expected order optimality. Then, we derive the expected gap between the traffic volume produced by scheme $\mathcal{F}_o$ and the cut-set bound under the specific cache size distribution. Finally, we further quantitatively dig the impact on the total traffic volume of GCD by developing a more tighter bound.

**A. Problem Statement**

Let $M_1 = \{M_{1,1}, M_{2,1}, \ldots, M_{K,1}\}$ denote the cache set, where $M_1$ is a set of order statistics that comes from a common parental distribution $F$. Here, the order statistics represent the a set of independent random variables such that $M_{1,1} \leq M_{2,1} \leq \cdots \leq M_{K,1}$. Further detail can be seen in [17].
A memory-traffic volume pair \((M, E[R_M])\) is achievable for scheme \(\mathcal{K}\) under requests \((d_1, \ldots, d_K)\) if every user \(k\) is able to reconstruct its requested content \(W_{dk}\) with error probability \(P_e \to 0\) and produce the expected traffic volume \(E[R_M(d_1, \ldots, d_K)]\) bits. Note that expectation is on the cache size. A memory-traffic volume pair \((M, E[R_M])\) is achievable for scheme \(\mathcal{K}\) if this pair is achievable for every possible requests \((d_1, \ldots, d_K)\) in the delivery phase. Defined by

**Definition 7**: (Expected achievable scheme)

\[
E[R_M] \triangleq \max_{(d_1, \ldots, d_K) \in N^K} E[R_M(d_1, \ldots, d_K)]
\]  
(34)

the expected worst case normalized traffic volume for scheme \(\mathcal{K}\).

We use the \(E[R^*]\) to represent the smallest traffic volume such that \((M, E[R^*])\) is achievable. Defined by

**Definition 8**: (Expected optimal scheme)

\[
(M, E[R^*]) \triangleq \inf \{ (M, E[R_M]) : \forall F, \mathcal{K} \}
\]  
(35)

the infimum of all achievable \((M, E[R_M])\).

Correspondingly, the order optimality is defined by

**Definition 9**: (Expected order optimal) The scheme \(\mathcal{K}\) is order optimal if only if

\[
E[R_M(M, K, N)] \leq C(F).
\]  
(36)

We can see that this constant is function of the parental distribution \(F\). In the following discussion, we will show that this constant is a function of 1st moment and 2nd moment of \(F\) when \(K = 2\).

**B. Analysis under Probabilistic Cache Set**

The following theorems use the simple case \(K = 2\) to illustrate the both constant gaps of coded caching scheme and uncoded version are related to the variance of the cache set.

**Theorem 6**: (Order optimality for normal distributed cache size) For the caching problem, \(N(N \geq 2)\) contents, \(K = 2\) users each with cache size \(M = \{M_{1,2}, M_{2,2}\}\), and comes from a normal distribution \(N(\mu, \sigma^2)\), then

\[
E[R_M(M, N, K)] \leq 2 - \frac{\sqrt{\pi} \mu + \sigma}{\sqrt{\pi} N}.
\]  
(37)

**Proof**: Based on the definition of expected traffic volume and cut-set bound, we have

\[
R_{\mathcal{K}_2}(M, N, 2) = \left(1 - \frac{M_{1,2}}{N}\right) + \left(1 - \frac{M_{2,2}}{N}\right)
\]  
(38)

\[
R^*(M, N, 2) \geq \left(1 - \frac{M_{1,2}}{N}\right).
\]  
(39)

Then,

\[
E\left[\frac{R_{\mathcal{K}_2}(M, N, 2)}{R^*(M, N, 2)}\right] \leq E\left[1 + \left(1 - \frac{M_{2,2}}{N}\right)\right] = 2 - \frac{E[M_{2,2}]}{N}.
\]

Using the results in [17] that \(E[M_{2,2}] = \mu + \sigma/\sqrt{\pi}\), we have

\[
E\left[\frac{R_{\mathcal{K}_2}(M, N, 2)}{R^*(M, N, 2)}\right] \leq 2 - \frac{\sqrt{\pi} \mu + \sigma}{\sqrt{\pi} N}.
\]

**Theorem 7**: (Gap of the uncoded caching scheme) For the caching problem, \(N(N \geq 2)\) contents, \(K = 2\) users each with cache size \(M = \{M_{1,2}, M_{2,2}\}\), and comes from a normal distribution \(N(\mu, \sigma^2)\), then

\[
E\left[\frac{2 - (M_{1,2} + M_{2,2})/N}{R^*(M, N, 2)}\right] \leq 2 - \frac{2\sigma}{\sqrt{\pi} N}.
\]  
(38)

**Proof**: Similarly, consider

\[
R^*(M, N, 2) \geq \left(1 - \frac{M_{1,2}}{N}\right),
\]

Then,

\[
E\left[\frac{2 - (M_{1,2} + M_{2,2})/N}{R^*(M, N, 2)}\right] \leq E\left[\frac{2 - M_{2,2} - M_{1,2}}{N - M_{1,2}}\right]
\]  
(39)

\[
< 2 - \frac{E[M_{2,2} - M_{1,2}]}{N}
\]

\[
= 2 - \frac{2\sigma}{\sqrt{\pi} N}.
\]

From Theorem 6 and 7 we can clearly see how the constant gap related to the variance of the cache set, where the variance of the cache set plays a linear negative role in such constant. And this gap has a faster speed in the uncoded cache, seen 2\(\sigma\) rather than the \(\sigma\) of coded version. Consider

\[
E\left[2 - \frac{M_{1,2} + M_{2,2}}{N}\right] = 2 - \frac{2\mu}{N},
\]

the traffic volume produce by the uncoded caching scheme is irrelevant to the variance of cache set. Thus the descending gap of uncoded version mainly come from the increase of the cut-set bound, which further implies the deviation of cache set will strengthen the inherent miss of coding opportunities.

Remark that the variance \(\sigma\) cannot be infinity since the range of \(M)\) is \([0, N]\).

In Theorem 4 we have answered our second question that what condition the cache set should satisfy such that the grouping of users will not improve the traffic volume compared to the non-grouping scheme. However, this condition is mostly strong, i.e., the cache size of almost all users should be zero, such that rarely happen in the real scenario. In fact, this question can be relaxed and turns into a trade-off problem between the increment of traffic volume due to the grouping and the deviation of cache set. In this case, previous condition can be regarded as an extreme case of this trade-off that the deviation is largest and traffic volume increment is zero.

Here, we use the concept of probabilistic cache set to develop an upper bound of the increment of traffic volume, which is much tighter than \(G_{\mathcal{K}_2}\) and related to the deviation of the cache set.

**Theorem 8**: (Expected miss coding ratio of GCD) For caching problem with \(N\) contents, \(K(K < N)\) users with
probabilistic cache set $M_K$, we divide all users into $L$ groups $\mathcal{K} = \{K_1, \ldots, K_L\}$, $M_K = \{M_{K_1}, \ldots, M_{K_L}\}$, then we have,

$$1 \leq \mathbb{E}[\mathfrak{m}_G(M_K, N, K, L)] \leq \mathbb{E}[H_{\tilde{\mathfrak{s}}, L}(M_K, N, K, L)].$$  \hspace{1cm} (39)

Fig. 4. The comparison between approximate upper bound (40) and real value of increment of traffic volume under four kinds of deviation of cache set. The system parameters $N = 500$, $K = 300$ and $\mu = 200$.

This kind of upper bound $\mathbb{E}[H_{\tilde{\mathfrak{s}}, L}(M_K, N, K)]$ is dependent on the structure of parental distribution $F$, the number of groups $L$, the number of contents $N$ and the number of users $K$. Due to analytically intractable of the close-form expression of $\mathbb{E}[H_{\tilde{\mathfrak{s}}, L}(M_K, N, K)]$, we use the numerical methods to study the likely form of it under probabilistic cache set. Denote by

$$\mathbb{E}[H_{\tilde{\mathfrak{s}}, L}(M_K, N, K)] \leq \frac{1}{1 + e^{-\frac{N\mu}{L}}} L^{\frac{\sigma}{\mu}}, L > 1.$$  \hspace{1cm} (40)

It can be seen that this bound is dependent on the expectation and variance of distribution $F$ and shows a trend that the increment of the traffic volume will be reduced when the variance increases. The reason that we get this kind of form of this upper bound can be seen in the full version of this paper [10]. The Fig. 3 partly shows the tightness and effectiveness of this bound.

As show in Fig. 4 when the variance is extremely small, our approximate bound is tight, especially when the number of groups is small, while when the variance is large, this approximate bound will be a little loose. Besides that, another important observation is that he increment of traffic volume is factually convergent to a constant. In particular, when the variance $\sigma = \mu$, this increment is bounded by only a constant factor $2$!

VI. NUMERICAL RESULTS

In the previous section, all analysis is in the scenario that $K \leq N$. In this section, we present the numerical results when $K > N$. Then, we investigate the impact of the system parameters on the delivery-phase traffic volume. Moreover, we systematically investigate the performance of GCD.

A. The Effect of Cache Set

We have proven that when $K = 2$, the constant gap is related to the variance of the cache set. For large $K$, we use the numerical analysis to show how the constant gap scales as the deviation of the cache set. To guarantee the deviation of the cache set, the average cache size cannot be too large. In fact, this constraint is reasonable, since the number of contents is mostly much larger than the cache size. Thus, we set the maximum average cache size $\mu = 0.3N$, and the cache set comes from a normal parental distribution.

The gap in Fig. 5 and Fig. 6 refers to the constant gap between traffic volume produced by scheme $\tilde{\mathfrak{s}}$, and cut-set bound. Fig. 5 plots the gap versus the average cache size $\mu$ under fixed variance of cache set. The variance under small and large deviation are denoted by $\sigma = 0.1\mu$ and $\sigma = 0.7\mu$, separately. The zero deviation refers to the traditional case that the cache size is uniform and we regard it as a baseline. In the Fig. 5(a), the number of contents $N = 100$ is larger than the number of users $K = 50$. The gap under large deviation is strictly less than that under small deviation, and less than the baseline. While in the Fig. 5(b) that $N = 50$, $K = 100$, this relation shows piecewise characteristics: when the average cache size is small ($\mu < 0.25N$), it has the same manner as $N > K$, when the average cache size is large ($\mu > 0.25N$), the gap under homogenous cache set decreases dramatically and less than the other two cases.

Fig. 5. The impact of average cache size on the gap between $R_{\tilde{\mathfrak{s}}}(M, N, K)$ and $R^o(M, N, K)$. Note that the scale in the (a) is a semi-logarithmic coordinate system.

The gap in Fig. 6 plots the gap versus the variance of the cache set. Note that the scale in the (a) is a semi-logarithmic coordinate system.

Fig. 6. The impact of deviation of cache set on the gap.
performance gain of coded cache compared with the uncoded cache is $15 \times$ under small deviation ($\sigma = 0.05\mu$), while only approximately $3 \times$ under large deviation ($\sigma = 0.6\mu$). For the case that $N > K$, seen in Fig. 6(a), both gap-variance curves of coded and uncoded cache have a declining trend and uncoded version performs faster. While for the case that $K < N$, seen in Fig. 6(b), the gap-variance curve of coded cache shows an unique single valley manner: under small cache sizes, it first increases, then it decreases linearly when the variance is large.

**B. The Impact of number of contents and users: $N$ and $K$**

Then we present how the gap scales as the system parameters such as $N$ and $K$. Assume there are two relationships between $K$ and $N$: $N = \Theta(K)$ and $N = \omega(K)$. The first case refers to the number of users and the number of contents have the same order. The second case refers to the number of content is extremely larger that the number of users. The characteristic of cache set is $\mu = 30, \sigma = 0.3\mu$.

**Fig. 7.** The impact of number of groups $L$, the variance $\sigma$ and the average cache size $\mu$ on the performance of GCD. System parameters: $N = 300, K = 100$.

In Fig. 7(a) and (b), we plot the gap versus the number of groups under four kinds of cache sets with increasing variance, it can be seen that the gap except the homogeneous cache set will be approximate to a constant when the number of groups increases, and this constant under small cache sized has a faster convergent rate than that under large cache size. Moreover, from Fig. 7(c) and (d), we can see that the gap decreases sharply when the variance of the cache set increases and larger average cache size will produce a faster decreasing rate. In the practical scenario, the average cache size $\mu = 0.1N$ and the variance is $\sigma = 0.5\mu$. If we divide the 300 users into 50 groups and each group has 6 users, the gap shown in the Fig. 7(a) is just 9 that is much less than our estimated bound $4 \cdot 50^{2/3} \approx 50$.

**Fig. 8.** The impact of system parameters $N$ and $K$ on the gap between $R_{\text{GCD}}(M, N, K)$ and $R^c(M, N, K)$. The average cache size is $\mu = 30$.

In Fig. 8(a), we can see that the gap gradually increases to a constant when $N$ is sufficiently large, while in Fig. 8(b), the gap gradually decreases to 1. Based on the asymptotic analysis, we can get easily get the reason for $N = \omega(K)$. Since $N \to \infty$, each term in (24) will approximate to 1 and $R_{\text{GCD}}(M, N, K)$ will approximate to $K$. In the similar manner, the cut-set bound $R^c(M, N, K)$ will also approximate to $K$. Thus the gap will approximate to 1. The reason for $N = \Theta(K)$ is based on an extremely complicated series analysis and can be seen in our full version paper [10].

**C. The Performance of GCD**

Here we investigate the performance of GCD compared with the cut-set bound $R^c$ when $N = 300, K = 100$. For the scenario that $N < K$, the results are similar and we omit this part. Eight kinds of cache set tested. The average cache size $\mu$ evaluates from $\{30, 150\}$ to denote the small and large cache size, separately. The variance of cache set $\sigma$ evaluates from $\{0, 0.25\mu, 0.5\mu, \mu\}$, to denote the increasing trend of deviation of the cache set. The results are shown in Fig. 7. Here the gap means the constant gap between the total traffic volume produced by GCD and the cut-set bound of the original system, and the partition of users is completely random.

In the Fig. 7(a) and (b), we plot the gap versus the number of groups under four kinds of cache sets with increasing variance, it can be seen that the gap except the homogeneous cache set is much smaller than our estimated bound $4 \cdot 50^{2/3} \approx 50$.

**VII. CONCLUSION AND EXTENSION**

In this paper, we have investigated the fundamental limits of coded cache under the heterogeneous cache set. Through deriving the cut-set bound in this regime, we have pointed out that even traditional coded caching scheme with zero-bit-paddung can guarantee the order optimality. Moreover, using the concept of probabilistic cache set, we have proven such gap is closely related to the deviation of cache set, i.e., the gap will decrease when the variance among users’ cache sizes increases. Besides that, we also have studied the group coded delivery scheme in this regime and presented that, although both heterogenous cache set and group coded delivery will lead to miss of coding opportunities, the combination of them will weaken this effect, even the group coded delivery still shows the constant gap the optimum when the cache set has the small deviation.

**A. Heterogenous Coded Delivery**

In this subsection, we discuss whether there exists an efficient scheme can counteract this "bit waste" phenomenon
and produce better performance. As analyzed in the previous section, this phenomenon is mainly caused by the different length of the segment in $V_{k,U/k}$. In fact, this phenomenon is also emerged in the regime of nonuniform populated contents. In this case, each content is allocated the different size of memory space and the segment $V_{k,U/k}$ will have the different length across different $k$, further details can be seen in [5].

To counteract this phenomenon, recent work [15] proposes a novel technique called heterogenous coded delivery (HCD), which is based on the following observation: the traditional decentralized coded caching scheme only consider the coding opportunities restricted to form cliques only between the segments of the same type and thus missed coding opportunities. They build upon this observation and design a new delivery scheme that considers more coding opportunities by forming cliques between segments not only of the same type, but also of different types. We incorporate this technique into our scenario and prove this technique cannot produce extra performance gain. We refer this scheme as $\mathcal{F}_h$. The pseudocode is listed in Algorithm 4.

Algorithm 4: Delivery phase in HCD scheme $\mathcal{F}_h$.

```plaintext
for (k = 1, k ≤ K; k++) do
  for choose k users from K users to form a subset U do
    Maxsize ← max$_{k\in U}|V_{k,U/{k}}|$
    for l ∈ U do
      if $|V_{l,U/l}| < Maxsize$ then
        temp ← (Maxsize − $|V_{l,U/l}|$) bits of all segments $V_{l,U'/l} : U \subset U'$;
        $V_{l,U/l}' ← V_{l,U/l} +$ temp;
        $X_U ← X_U \oplus V_{l,U/l}'$;
    Multicast the coded data $X_U$ to users in $U$.
```

It can be seen that the core of this scheme is to pad the bits coming from the higher type segments instead of useless zero bits to the lower type segments that has the shorter length.

**Theorem 9:** The HCD scheme $\mathcal{F}_h$ produced the same traffic volume as scheme $\mathcal{F}_o$, denote by

$$R_{\mathcal{F}_h}(M,N,K) = R_{\mathcal{F}_o}(M,N,K).$$

Proof: To prove the equivalence of the traffic volume produced by these two schemes, we only need to prove the higher-bits-padding operation in step $\forall i \in K$ and user group $\forall U, |U| = i$ will not reduce the signal size $X_U$, for $\forall U', U \subset U'$. Based on the Lemma [1] that segment size is ordered, we have

$$|X_U| = \max_{k\in U}|V_{k,U/{k}}| = |V_{\inf U,U/\{\inf U}}|.$$

For user $m \in U, m > \inf U$, the size of its segment $V_{m,U/\{m\}}$ is less than $X_U$ and should be higher-bits-padded by all non-empty segments $V_{m,U'/\{m\}}, U \subset U'$. Since $U \subset U'$, the $k > \inf U > \inf U'$, thus we have

$$|X_U'| = |V_{\inf U',U'/\{\inf U'}}| = \max_{k\in U'}|V_{k,U/{k}}| > |V_{m,U'/\{m\}}|,$$

which means that although the operation in step $i$ for user group $U$ reduce the size of segment $|V_{m,U'/\{m\}}|$, it cannot reduce the size of the signal $X_{U'}$.

The above theorem utilizes the property of ordered signal size to prove that the HCD scheme produced the same traffic volume. From the procedure of the proof, we can see that the although padding the bits coming from the higher type segments can reduce the size of them, this kind of reducing effect never happens in the longest segment in each user group, thus it cannot reduce the traffic volume.

**B. Coded Cache under Heterogenous Cache Set and Random User Demands**

As we discussed before, the miss of coding opportunities not only occurs in the coded cache under heterogenous cache set, but also under the nonuniform populated contents or random user demands. Similarly, an open question is, if we consider random user demands under the heterogenous cache set, whether such miss of coding phenomenon will be strengthened. This line of work is further discussed in [13].

**C. Centralized Coded Cache Under Heterogenous Cache Set**

In this paper, we mainly focus on the decentralized coded caching scheme in [3]. This scheme does not require a centralized control in the placement phase and the random caching procedure in the placement phase guarantees that the division of each content is “automatic” and the size of each segment is determined by the size of cache that storing them. While in the centralized coded cache, the division of each content is pre-defined. For the homogeneous cache set $\mathcal{M}$, each content is divided into $\frac{K}{t}$ equal segments, where $t = K\overline{M}/N$. However, this kind of division cannot applied straightforwardly to the heterogenous cache set. Since the cache size of each user is different, the number of segments of each content being divided will be bound by the smallest cache size $K\overline{M}_1/N$. In fact, we can divide each content unequally. We use the following example to illustrate our main idea.

**Example 3** (Centralized coded cache under heterogenous cache set.) Consider $N = 3, K = 3$ so that there are three contents $W_1, W_2, W_3$. Assume the ordered cache set $\mathcal{M} = \{M_1, M_2, M_3\}$ satisfying $\overline{M} = 2$. In the placement phase, we split each content into 3 segments, namely $W_i = (W_{i,12}, W_{i,13}, W_{i,23})$, and the caching strategy is

$$Z_1 = (W_{1,12}, W_{1,13}, W_{2,12}, W_{2,13}, W_{3,12}, W_{3,13});$$

$$Z_2 = (W_{1,12}, W_{1,23}, W_{2,12}, W_{2,23}, W_{3,12}, W_{3,23});$$

$$Z_3 = (W_{1,23}, W_{1,13}, W_{2,23}, W_{2,13}, W_{3,23}, W_{3,13}).$$

Assume the content $W_{i,12}, W_{i,13}$ and $W_{i,23}$ occupies the $\alpha, \beta$ and $\gamma$ percent of each content. Then we have the following
equations that coming from the cache size constraint.

\[
\begin{aligned}
3(\alpha+\beta) &= M_1 \\
3(\alpha+\gamma) &= M_2 \\
3(\beta+\gamma) &= M_3 \\
\alpha+\beta+\gamma &= 1
\end{aligned}
\]

Solving above overdetermined equation, we can get the size of each segment and the traffic volume is

\[
1 - \frac{M_1}{3}
\]  \hspace{1cm} (42)

From (42), we can see that the traffic volume is determined by the smallest cache size, which is identical to the cut-set bound. Here, we may have a question that if the centralized code cache scheme is order optimal under heterogeneous cache set. This is discussed further in [11].

D. Group Coded Delivery

Although GCD will increase the traffic volume, the previous analysis shows that, when the users’ cache sizes imply large difference, random partition of users still can guarantee the order optimality. This result is meaningful, since the GCD has many advantages in the practical scenario:

1) Decrease the transmission complexity. For a coded caching system that has 100 users, the transmission complexity is \(2^{100} \approx 10^{30}\). If we divide 100 users into 20 groups with each group having 5 users, the transmission complexity is only \(20 \cdot 2^5 \approx 10^3\), which is much less than the original system.

2) Improve the performance of the wireless fading channel. In this case, the shared link is not error-free, instead, each user will experience different channel condition and there exists a phenomenon called “multicast saturation” that the multicast capacity will be bounded by the user that has the worst channel condition. The recent work [14] shows that the coded caching scheme has an extremely limited performance gain, sometimes even worse than the uncoded cache. However, through grouping the users based on their channel condition, the performance can be further improved.

One promising direction is to prove that for any cache set \(M_K\), there exists maximal \(L_m\) that can still guarantee the order optimality. If such \(L_m = \Theta(K)\), the complexity of coded caching scheme can reduced to the linear complexity and still order optimal. This line of work is not our focus, here we consider the raw version of the GCD that dividing \(K\) users into \(L\) groups, and the following example shows our motivation.

Example 4 (Optimal Group Coded Delivery.) Consider another simple system distributing \(N\) contents to \(K = 4\) users, the ordered cache set is \(M = \{M_1, M_2, M_3, M_4\}\). It is required to divide the users into \(L = 2\) groups. The total number of partitions is 7, seen in TABLE II.

Remark that \(\bar{m}_i = 1 - \frac{M_i}{M}\). Apparently, we can get the traffic volume of \(D_1 < D_2\) and \(D_4 < D_3\), while find the minimal traffic volume is impossible without giving the value of cache set \(M\). In fact, finding the optimal \(L\)-partition that produces the lowest traffic volume in this regime is nontrivial and shows the NP-hardness in the strong sense. The further discussion can be seen in [12].

E. Application of Equivalent Network Diagram

In the Section IV, we have shown that the coded caching scheme under broadcast network is equivalent to a simple unicast system with hierarchical cache access. Here we use this equivalent network diagram to prove that the finding the optimal partition of users has a trivial solution when the cache set is homogeneous.

![Fig. 9. Application of equivalent network diagram in group coded cache.](image)

As can be seen in Fig 9, the partition of users in the group coded delivery is equal to break the connection among caches in the different groups of our equivalent network diagram. If the cache set is homogenous, the capacity of different connections are identical. Therefore, finding the optimal \(L\)-partition is equal to find \(L-1\) cut points that break minimal connections in our equivalent network diagram. According to the symmetry of our equivalent network diagram, the optimal \(L - 1\) should be the first \(L - 1\) points, which means the optimal partition is assigning only one user into first \(L - 1\) group, then assign the rest \(K - L + 1\) users to the last group. In fact, in the regime of heterogeneous cache set, this network diagram can also be utilized to design the high-efficient approximate algorithm to find the optimal \(L\)-partition.

References

[1] Cisco, “Cisco visual networking index: Global mobile data traffic forecast update 2012-2017,” White paper, 2013.
[2] M. A. Maddah-Ali, U. Niesen, “Fundamental limits of caching,” in IEEE Transactions on Information Theory, vol. 60, no.5, pp. 2856C-2867, 2014.
[3] M. A. Maddah-Ali, U. Niesen U, “Decentralized coded caching attains order-optimal memory-rate tradeoff,” IEEE/ACM Transactions on Networking, vol. PP, no.1, pp. 1, 2014.
[4] U. Niesen and M. A. Maddah-Ali, “Coded caching with nonuniform demands,” in Proc. IEEE INFOCOM WSKHPS, 2014.
[5] S. Wang, X. Tian and H. Liu, “Exploiting the Unexploited of Coded Caching for Wireless Content Distribution,” in Proc. IEEE ICNC, 2015.
Then we have $M_n < M_m$, this is contradictory to the definition of subset $V$. Hence
\[
\sum_{i \in V} M_i \leq sM, \tag{51}
\]
and this guarantee (47).

\[\square\]

B. Proof of Lemma 2

Proof: We prove this Lemma by naturally summarizing the traffic volume produced in each transmission. Remark following a series of operations (b) is based on Lemma 1 on the properties of ordered signal size. Based on the delivery procedure of Algorithm 2, we have,

1) When $k = K$, the multicasting user group $U = K$, and the size of the transmission signal $X_K$ is,
\[
|X_K| = |V_{1,K/(1)}| = \left(1 - \frac{M_1}{N}\right) \prod_{i=2}^{K} \left(\frac{M_i}{N}\right) = P_1Q_1; \tag{52}
\]

2) When $k = K - 1$, there are $\binom{K}{1}$ multicasting user groups $U$.
If $U = K/\{i\}, 1 < i \leq K$, the size of $X_U$ is,
\[
|X_U| \overset{(b)}{=} \left|V_{U/(1)}\right| = \left(1 - \frac{M_1}{N}\right) \left(1 - \frac{M_i}{N}\right) \prod_{j=2,j\neq i}^{K} \left(\frac{M_j}{N}\right) = P_1Q_1 \left(\frac{N - M_i}{M_i}\right);
\]
If $U = K/\{1\}$, the size of $X_U$ is,
\[
|X_U| \overset{(b)}{=} \left|V_{2,U/(1)}\right| = \left(1 - \frac{M_1}{N}\right) \left(1 - \frac{M_2}{N}\right) \prod_{j=3}^{K} \left(\frac{M_j}{N}\right) = P_2Q_2;
\]
Thus, when the multicasting group has $(K - 1)$ users, the total size of transmission signal is
\[
\sum_{U \subseteq \mathcal{K}, |U| = K - 1} |X_U| = P_1Q_1 \sum_{i=2}^{K} \left(\frac{N - M_i}{M_i}\right) + P_2Q_2 = P_1Q_1S_1 + P_2Q_2.
\]

3) When $k = K - 1$, there are $\binom{K}{2}$ multicasting user groups $U$.
If $U = K/\{i,j\}, 1 < i < j \leq K$, the size of $X_U$ is,
\[
|X_U| \overset{(b)}{=} \left|V_{1,U/(1)}\right| = \left(1 - \frac{M_1}{N}\right) \left(1 - \frac{M_i}{N}\right) \left(1 - \frac{M_j}{N}\right) \prod_{k=2,k\neq i,j}^{K} \left(\frac{M_k}{N}\right)
\]
\[ = \left(1 - \frac{M_i}{N}\right) \left(\frac{N-M_i}{N}\right) \left(\frac{N-M_j}{N}\right) \prod_{k=2}^{K} \left(\frac{M_k}{N}\right) \]
\[ = P_1 Q_1 \left(\frac{N-M_i}{M_i}\right) \left(\frac{N-M_j}{N}\right) ; \]
If \( U = K/\{i, j\} \), \( i = 1, 2 < j \leq K \), the size of \( X_U \) is,
\[ |X_U|_{(b)} = |V_{2, U}/(2)| = \left(1 - \frac{M_i}{N}\right) \left(1 - \frac{M_j}{N}\right) \prod_{k=3, k \neq j}^{K} \left(\frac{M_k}{N}\right) \]
\[ = \left(1 - \frac{M_i}{N}\right) \left(1 - \frac{M_j}{N}\right) \prod_{k=3}^{K} \left(\frac{M_k}{N}\right) \]
\[ = P_2 Q_2 \left(\frac{N-M_i}{N}\right) ; \]
If \( U = K/\{i, j\} \), \( i = 1, j = 2 \), the size of \( X_U \) is,
\[ |X_U|_{(b)} = |V_{3, U}/(3)| = \left(1 - \frac{M_i}{N}\right) \left(1 - \frac{M_j}{N}\right) \left(1 - \frac{M_3}{N}\right) \prod_{k=4}^{K} \left(\frac{M_k}{N}\right) \]
\[ = P_1 Q_1 \left( S_1 \bigotimes S_2 \right) + P_2 Q_2 S_3 + P_3 Q_3. \]

4) \ldots \ldots \]
5) When \( k = 1 \), there are \( \binom{K}{1} \) multicasting user groups \( U \).
If \( U = \{i\}, 1 < i \leq K \), the size of \( X_U \) is,
\[ |X_U|_{(b)} = |V_{1, 0}| = \prod_{i=1}^{K-1} \left(1 - \frac{M_i}{N}\right) = P_1 Q_1 \left( \bigotimes_{K-i=1}^{K} S_{i+1} \right) ; \]
Thus, when the multicasting group has only one user, the total size of transmission signals is
\[ \sum_{|U|=1} |X_U| = P_1 Q_1 \sum_{1<i<j \leq K} \left(\frac{N-M_i}{M_i}\right) + P_2 Q_2 \sum_{i=2}^{K} \left(\frac{N-M_i}{M_i}\right) + P_3 Q_3 \]
\[ = P_1 Q_1 \left( S_1 \bigotimes S_2 \right) + P_2 Q_2 S_3 + P_3 Q_3. \]

C. Proof of Corollary 2

Proof:
\[ R(M, N, K) = \sum_{i=1}^{K} \left(1 - \frac{M}{N}\right)^i \]
\[ = \left(1 - \frac{M}{N}\right) - \left(1 - \frac{M}{N}\right)^{K} \]
\[ = K \left(1 - \frac{M}{N}\right) \frac{N}{KM} \left[1 - \left(1 - \frac{M}{N}\right)^K \right] . \]
The proof of order optimality is extremely tedious and the complete procedure can be seen in the [3].

D. Proof of Corollary 3

Proof:
\[ R_{\hat{\delta}_0}(M_u, N, K) \]
\[ = \sum_{i=1}^{K-1} \left(1 - \frac{M}{N}\right)^i + \left(1 - \frac{(1+\alpha)M}{N}\right) \left(1 - \frac{M}{N}\right)^{K-1} \]
\[ = R_{\hat{\delta}_0}(M_u, N, K) + \left(1 - \frac{(1+\alpha)M}{N}\right) \left(1 - \frac{M}{N}\right)^{K-1} \]
\[ = \left(1 - \frac{M}{N}\right)^{K} \]
\[ = R_{\hat{\delta}_0}(M_u, N, K) - \left(1 - \frac{M}{N}\right)^{K-1} . \]

E. Proof of Corollary 4

Proof:
\[ R_{\hat{\delta}_0}(M_d, N, K) \]
\[ = \sum_{i=1}^{K} \left(1 - \frac{M}{N}\right)^i + \left(1 - \frac{M}{N}\right)^{K} \sum_{i=\frac{K}{2}+1}^{K} \left(1 - \frac{\alpha M}{N}\right)^{i-\frac{K}{2}} \]
\[ = R_{\hat{\delta}_0}(M_1, N, K) + \left(1 - \frac{M}{N}\right)^{K} R_{\hat{\delta}_0}(M_2, N, K) . \]

F. Proof of Corollary 5

Proof: For simplicity, we substitute \( 1 - M_i/N \) by \( \tilde{m}_i \), and \( 1 - M/N \) by \( \tilde{m}_M \). We use the function \( f(\tilde{m}_1, \ldots, \tilde{m}_K) \) to denote the traffic volume formula \( R_{\hat{\delta}_0}(M, N, K) \). Then, we transform the Theorem 5 to the following problem that finding the minima of a function subject to equality constraints.
\[ \min f(\tilde{m}_1, \ldots, \tilde{m}_K) = \sum_{i=1}^{K} \left(\prod_{j=1}^{i} \tilde{m}_j \right) ; \]
\[ \text{s.t} \sum_{i=1}^{K} \tilde{m}_i = K \tilde{m}, \tilde{m}_i \geq \tilde{m}_j, \forall i < j . \]
We use the following procedure to prove.
First, we prove the function $f(\tilde{m}_1, \ldots, \tilde{m}_K)$ is strictly decreasing under the following operation $\delta$: when $\tilde{m}_k \neq \tilde{m}_{k+1}$, replace them by $\tilde{m}_k' = \tilde{m}_{k+1} = (\tilde{m}_k + \tilde{m}_{k+1})/2$, $0 < k < K$, denote by

$$\Delta_\delta = f(\tilde{m}_1, \ldots, \tilde{m}_k, \tilde{m}_k', \tilde{m}_{k+1}, \ldots, \tilde{m}_K) - f(\tilde{m}_1, \ldots, \tilde{m}_k, \tilde{m}_{k+1}, \ldots, \tilde{m}_K) > 0.$$  

Note that this operation satisfies the constraint \[56\].

Second, we use the contradictory argument to prove that function reaches the minima only if $\tilde{m}_1 = \tilde{m}_2 = \cdots = \tilde{m}_K$. Assume the function $f(\tilde{m}_1, \ldots, \tilde{m}_K)$ reaches the minima in $(\tilde{m}_1^*, \ldots, \tilde{m}_K^*)$. If $\forall r \in K$ such that $\tilde{m}_r^* \neq \tilde{m}_r^{*+1}$, then we adopt the operation that $\tilde{m}_r^* = \tilde{m}_r^{*+1} = (\tilde{m}_r^* + \tilde{m}_r^{*+1})/2$ and the value of function $f$ will strictly decrease, which is contradictory to the out assumption that function $f$ reaches the minima in the point $(\tilde{m}_1^*, \ldots, \tilde{m}_K^*)$.

Then we prove the first argument. Based on the definition of function $f$, seen in \[55\], we have

$$\Delta_\delta = \left(\frac{\tilde{m}_k - \tilde{m}_{k+1}}{2}\right) \prod_{i=1}^{k-1} \tilde{m}_i + \tilde{D}_k \cdot (\tilde{m}_k \tilde{m}_{k+1} - \tilde{m}_k' \tilde{m}_{k+1}') / (\tilde{m}_k \tilde{m}_{k+1}) .$$

where

$$\tilde{D}_k = \prod_{i=1}^{k-1} \tilde{m}_i + \left(\sum_{i=k+2}^{K} \prod_{j=1}^{i} \tilde{m}_i\right) / (\tilde{m}_k \tilde{m}_{k+1}) .$$

Then,

$$\Delta_\delta = \left(\frac{\tilde{m}_k - \tilde{m}_{k+1}}{2}\right) \prod_{i=1}^{k-1} \tilde{m}_i - \tilde{D}_k \cdot \left(\frac{(\tilde{m}_k - \tilde{m}_{k+1})^2}{4}\right) = \left(\frac{\tilde{m}_k - \tilde{m}_{k+1}}{2}\right) \cdot \frac{\tilde{D}_k}{2} \left[2 \prod_{i=1}^{k-1} \tilde{m}_i - (\tilde{m}_k - \tilde{m}_{k+1})\right] .$$

Base on the definition of operation $\delta$, we have $\tilde{m}_k - \tilde{m}_{k+1} > 0$ and

$$\Delta_\delta > 0 \Leftrightarrow 2 \prod_{i=1}^{k-1} \tilde{m}_i - (\tilde{m}_k - \tilde{m}_{k+1}) > 0 .$$

Since

$$\frac{2 \prod_{i=1}^{k-1} \tilde{m}_i}{\tilde{D}_k} - (\tilde{m}_k - \tilde{m}_{k+1})$$

$$= \frac{2}{1 + \sum_{i=2}^{K-k} \prod_{j=2}^{i} \tilde{m}_k + \tilde{m}_{k+1}} - (\tilde{m}_k - \tilde{m}_{k+1})$$

$$\geq \frac{2}{\sum_{i=0}^{K-k-1} \tilde{m}_k + \tilde{m}_{k+2}} - (\tilde{m}_k - \tilde{m}_{k+1})$$

$$\geq \frac{2(1 - \tilde{m}_k + \tilde{m}_{k+1})}{\sum_{i=0}^{K-k+2} \tilde{m}_k + \tilde{m}_{k+2}} - (\tilde{m}_k - \tilde{m}_{k+1})$$

$$\geq 2(\tilde{m}_k - \tilde{m}_{k+1})/(\tilde{m}_k + \tilde{m}_{k+1}) - (\tilde{m}_k - \tilde{m}_{k+1})$$

$$\geq 2(\tilde{m}_{k+1} - \tilde{m}_k) - (\tilde{m}_k - \tilde{m}_{k+1})$$

$$\geq 2(\tilde{m}_{k+1} - \tilde{m}_k) - (\tilde{m}_k - \tilde{m}_{k+1})$$

$$\geq 0 .$$

Note that the left side in step (a) is strictly larger than the right side. Thus, the function $f$ is strictly monotonically decreasing of operation $\delta$.

G. Proof of Theorem \[4\]

Proof: We prove the first inequality.

$$\sum_{l=1}^{L} R(M_l, N, K) = \sum_{l=1}^{L} \sum_{i=1}^{K} \prod_{j=1}^{i} \left(1 - \frac{M_{\pi(l,j)}}{N}\right) \geq \sum_{l=1}^{L} \sum_{i=1}^{K} \prod_{j=1}^{i} \left(1 - \frac{M_j}{N}\right) = R(M, N, K) .$$

where $\pi(l,j)$ denotes the user identity of $j$th user in $l$th user group. (a) is based on the fact that $\pi(l,j) \geq j$. (b) is based on the fact that $K = \{\pi(l,j) : \forall l, j\}$. Considering the additive term in (a), we can verify the condition of equalization.

