A New Order-Optimal Decentralized Coded Caching Scheme with Good Performance in the Finite File Size Regime

Sian Jin, *Student Member, IEEE*, Ying Cui, *Member, IEEE*,
Hui Liu, *Fellow, IEEE*, Giuseppe Caire, *Fellow, IEEE*

**Abstract**

The decentralized random coded caching scheme proposed by Maddah-Ali and Niesen for the shared bottleneck link network achieves an order-optimal memory-load tradeoff when the file size goes to infinity. It was then successively shown by Shanmugam *et al.* that, in the practical operating regime where the file size is limited, such scheme yields a much less attractive worst-case coded caching gain. In this paper, we propose a decentralized random coded caching scheme that achieves a significantly larger worst-case coded caching gain than Maddah-Ali–Niesen’s decentralized random coded caching scheme in the finite file size regime, while its memory-load tradeoff is still order-optimal when the file size grows to infinity. We first propose a decentralized random coded caching scheme, aiming at ensuring abundant coded-multicasting opportunities when the file size is finite. Then, we analyze the worst-case load of the proposed scheme and show that it outperforms Maddah-Ali–Niesen’s and Shanmugam *et al.*’s decentralized schemes for finite file size, when the number of users is sufficiently large. Next, we analyze the asymptotic worst-case load of the proposed scheme when the file size goes to infinity and show that the proposed scheme achieves an order-optimal memory-load tradeoff. Finally, we characterize analytically the behavior of the worst-case coded caching gain of the proposed scheme as a function of the required file size, when the file size is large.

**Index Terms**

Coded caching, coded multicasting, content distribution, finite file size analysis.

S. Jin, Y. Cui and H. Liu are with Shanghai Jiao Tong University, China. G. Caire is with Technical University of Berlin, Germany. This paper was presented in part at IEEE GLOBECOM 2016.
I. INTRODUCTION

The rapid proliferation of smart mobile devices has triggered an unprecedented growth of the global mobile data traffic, with a predicted nearly seven-fold increase between 2016 and 2021 [1]. In order to support such dramatic growth of wireless data traffic, caching and multicasting have been recently proposed as two promising approaches for massive content delivery in wireless networks. By proactively placing content closer to or even at end-users during the off-peak time, network congestion during the demand peaks can be greatly reduced [2]–[4]. On the other hand, leveraging the broadcast nature of the wireless medium by multicast transmission, popular content can be delivered to multiple requesters simultaneously [5], [6].

Note that in [2]–[5], caching and multicasting are considered separately. In view of the benefits of caching and multicasting, joint design of the two promising techniques is expected to achieve superior performance for massive content delivery in wireless networks. For example, in [7], the optimization of caching and multicasting, which is NP-hard, is considered in a small cell network, and a simplified solution with approximation guarantee is proposed. In [8], the authors propose a joint throughput-optimal caching and multicasting algorithm to maximize the service rate in a multi-cell network. In [9]–[11], the authors consider the analysis and optimization of caching and multicasting in large-scale wireless networks modeled using stochastic geometry. However, [7]–[11] only consider joint design of traditional uncoded caching and multicasting, the gain of which mainly derives from making content available locally and serving multiple requests of the same contents concurrently.

Recently, a new class of caching schemes, referred to as coded caching, have received significant interest, as they can achieve order-optimal memory-load tradeoffs through wise design of content placement in the user caches. The main novelty of such schemes with respect to conventional approaches (e.g., as currently used in content delivery networks) is that the messages stored in the user caches are treated as “receiver side information” in order to enable network-coded multicasting, such that a single multicast codeword is useful to a large number of users, even though they are not requesting the same content. In [12], Maddah-Ali and Niesen consider a system with one server connected through a shared error-free bottleneck link to $L$ users. The server has a database of $N$ files, and each user has an isolated cache memory containing up to $M$ files. They formulate a caching problem consisting of two phases, namely, a content placement phase and a content delivery phase. The content placement is performed once, before operating the network, and
independently of the user demands. Then, users place demands in rounds, and at each round the server responds with a multicast message constructed by coded multicast XOR operations that satisfies all user demands simultaneously. The performance of the scheme is quantified by the length of the multicast message normalized by the size of a single file, i.e., by the load of the shared bottleneck link. This scheme has been successively investigated in a large number of recent works [12]–[24] under the same network topology and similar settings.

In [12], [13], the authors design centralized coded caching schemes to reduce the worst-case (over all possible requests) load of the shared link in the delivery phase. Specifically, Maddah-Ali–Niesen’s centralized scheme in [12] achieves an order-optimal memory-load tradeoff with the required file size (i.e., the number of packets per file, also referred to as the subpacketization level) greater than or equal to \((L^M_N)\). To reduce the required file size, Shanmugam et al.’s centralized scheme in [13] divides the \(L\) users into groups of size \(K\), and applies Maddah-Ali–Niesen’s centralized coded caching scheme for each user group separately. The required file size is greater than or equal to \((K^M_N)\), which is much smaller than that of Maddah-Ali–Niesen’s centralized scheme, at the sacrifice of worst-case load increase. The centralized schemes in [12], [13] have limited practical applicability since they require a centrally coordinated placement phase depending on the knowledge of the number of active users in the delivery phase, which is in practice not known in the placement phase.

In [13], [14], [16], the authors design decentralized coded caching schemes whose placement does not require any coordination and does not rely on the knowledge of the number of active users in the delivery phase, to reduce the worst-case load of the shared link in the delivery phase. Specifically, in [14], Maddah-Ali and Niesen propose a decentralized random coded caching scheme, which achieves an order-optimal memory-load tradeoff in the asymptotic regime of infinite file size. However, it was successively shown in [13] that this decentralized coded caching scheme can achieve at most a worst-case coded caching gain of 2 over conventional uncoded caching, if the file size is less than or equal to \(N^{LM} \exp \left( LM/N \right)\), which is referred to as the finite file size problem. The main reason for this negative result is that the random placement procedure in Maddah-Ali–Niesen’s decentralized scheme generates packets of random

\[1\] For future reference, in this paper, we refer to “worst-case coded caching gain” of a particular coded caching scheme as the ratio between the worst-case load achieved by conventional uncoded caching and the worst-case load achieved by that particular scheme. Since coding should provide a lower load, the gain is some number larger than 1.
lengths and causes large variance of the lengths of packets in each coded multicast message, leading to the “bit waste” effect in XOR operations of the delivery procedure [15] and a drastic reduction of coded-multicasting opportunities. To address the finite file size problem in decentralized coded caching, the authors in [16] propose two delivery procedures under the random placement procedure of Maddah-Ali–Niesen’s decentralized scheme [14]. Specifically, the first procedure in [16] is based on heterogenous coded delivery (HCD) [15], i.e., all packets (of the same type\(^2\)) in one coded multicast message are padded with data removed from some packets of higher types to achieve the same length as the longest packet in the coded multicast message. The second procedure in [16] is based on cross-type coded delivery, i.e., packets of different types are coded together, to increase the overall coded-multicasting opportunities. Note that the two delivery procedures could not fundamentally avoid the “bit waste” effect stemming from the random placement procedure of Maddah-Ali–Niesen’s decentralized scheme [14]. In [13], Shanmugam et al. propose a decentralized random coded caching scheme to reduce the “bit waste” effect when the file size is limited. The placement procedure partitions users into different groups and adopts the random placement procedure of Maddah-Ali–Niesen’s decentralized scheme for each group. As the user grouping method reduces the variance of lengths of packets caused by Maddah-Ali–Niesen’s random placement procedure, it is able to alleviate (although not fully avoiding) the “bit waste” effect. The delivery procedure includes a “pull down phase” based on a random appending method before the delivery procedure of Maddah-Ali–Niesen’s decentralized scheme. In the “pull down phase”, all packets of a selected type are randomly padded with data of higher types, to increase coded-multicasting opportunities in coding packets of the selected type at the expense of coded-multicasting opportunities in coding packets of higher types.

In [17]–[23], the authors design decentralized coded caching schemes to reduce the average (over random requests) load of the shared link in the delivery phase under an arbitrary file popularity. Specifically, in [19]–[21], Maddah-Ali–Niesen’s decentralized scheme [14] is extended in order to reduce the average load for an arbitrary but known in advance file popularity distribution. In [22], Maddah-Ali–Niesen’s decentralized scheme [14] is extended to the case of an unknown popularity distribution by placing contents on the fly. Note that the decentralized random coded caching schemes in [19]–[22] suffer from the same drawback as Maddah-Ali–Niesen’s decentralized scheme in [14] when the file size is limited. In [23], the authors propose

\(^2\)If a packet is stored in \( p \) caches of users, then the packet is said to be of type \( p \) [15].
a decentralized random coded caching scheme based on a greedy constrained coloring (GCC) delivery procedure. Compared with the delivery procedure of Maddah-Ali–Niesen’s decentralized scheme [14], the GCC delivery procedure has low computational complexity, but achieves the same average load [24]. The scheme based on the GCC delivery procedure in [23] can achieve an order-optimal load in the asymptotic regime of infinite file size with manageable complexity, but again suffers from the finite file size problem. In [17], [18], the authors propose decentralized random coded caching schemes based on cross-type coded delivery. Compared with the delivery procedure of Maddah-Ali–Niesen’s decentralized scheme [14], the cross-type coded delivery procedures can increase the overall coded-multicasting opportunities. The two schemes in [17], [18] are shown to achieve lower average loads when the file size is finite, but cannot fundamentally avoid the “bit waste” effect stemming from their random placement procedures.

In this paper, we consider the same problem setting as in [14], and propose a new decentralized random coded caching scheme. We focus on reducing the worst-case load of the shared link when the file size is finite, while maintaining an order-optimal memory-load tradeoff when the file size grows to infinity. Our main contributions are summarized below.

- Motivated by the content placement of Maddah-Ali–Niesen’s and Shanmugam et al.’s centralized schemes, we propose a placement procedure based on forming a collection of carefully pre-designed cache contents from which the users randomly and independently choose their cache contents. Such collection is referred to as the cache content base. This ensures zero variance of the lengths of packets involved in coded multicast XOR operations, hence fundamentally avoiding the “bit waste” effect and increasing coded-multicasting opportunities in the finite file size regime. Furthermore, we propose a delivery procedure which fully exploits coded-multicasting opportunities.
- We analyze the worst-case load achieved by the proposed scheme and show that the proposed scheme outperforms Maddah-Ali–Niesen’s and Shanmugam et al.’s decentralized schemes in the finite file size regime, when the number of users is sufficiently large.
- We analyze the asymptotic worst-case load of the proposed scheme when the file size goes to infinity and show that the proposed scheme achieves the same asymptotic memory-load tradeoff as Maddah-Ali–Niesen’s decentralized scheme, and hence is also order-optimal in the memory-load tradeoff.
- We analyze the worst-case coded caching gain of the proposed scheme in the finite file size regime. We show that when the number of users is large, the proposed scheme achieves the same tradeoff
between the required file size and the coded caching gain as Shanmugam et al.’s centralized coded caching scheme [13]. We derive an upper bound on the required file size for given target worst-case coded caching gain. We also analyze the growth of the worst-case coded caching gain with respect to the required file size of the proposed scheme, when the file size is large.

- Numerical results show that the proposed scheme outperforms Maddah-Ali–Niesen’s and Shanmugam et al.’s decentralized schemes in the finite file size regime, when the number of users is sufficiently large.

II. PROBLEM SETTING

As in [14], we consider a system with one server connected through a shared, error-free link to \( L \in \mathbb{N} \) users, where \( \mathbb{N} \) denotes the set of natural numbers. The server has access to a database of \( N \in \mathbb{N} \) (\( N \geq L \)) files, denoted by \( W_1, \ldots, W_N \), consisting of \( F \in \mathbb{N} \) indivisible data units.\(^3\) The parameter \( F \) indicates the maximum number of packets in which a file can be divided. Let \( \mathcal{N} \triangleq \{1, 2, \ldots, N\} \) and \( \mathcal{L} \triangleq \{1, 2, \ldots L\} \) denote the set of file indices and the set of user indices, respectively. Each user has an isolated cache memory of \( MF \) data units, for some \( M \in [0, N] \). We refer to \( \gamma \triangleq \frac{M}{N} \) as the normalized local cache size.

The system operates in two phases, i.e., a placement phase and a delivery phase [14]. In the placement phase, the users are given access to the entire database of \( N \) files. Each user fills its cache by using the database. Let \( \phi_l \) denote the caching function for user \( l \), which maps the files \( W_1, \ldots, W_N \) into the cache content \( Z_l \triangleq \phi_l(W_1, \ldots, W_N) \) for user \( l \in \mathcal{L} \). Let \( \phi \triangleq (\phi_1, \ldots, \phi_L) \) denote the caching functions of all the \( L \) users. Note that \( Z_l \) is of size \( MF \) data units. Let \( Z \triangleq (Z_1, \ldots, Z_L) \) denote the collection of cache contents of the \( L \) users. In the delivery phase, each user requests one file (not necessarily distinct) in the database. Let \( d_l \in \mathcal{N} \) denote the index of the file requested by user \( l \in \mathcal{L} \), and let \( d \triangleq (d_1, \ldots, d_L) \in \mathcal{N}^L \) denote the vector of requests of the \( L \) users. The server replies to these \( L \) requests by sending a multicast message over the shared link, received by all \( L \) users. Let \( \psi \) denote the server encoding function, which maps the files \( W_1, \ldots, W_N \), the cache contents \( Z \), and the requests \( d \) into the multicast message \( Y \triangleq \psi(W_1, \ldots, W_N, Z, d) \) sent by the server over the shared link. Let \( \mu_l \) denote the decoding function at user \( l \), which maps the received multicast message \( Y \), the cache content \( Z_l \), and the request \( d_l \), to the estimate \( \hat{W}_{d_l} \triangleq \mu_l(Y, Z_l, d_l) \) of the

\(^3\)The indivisible data units may be “bits” or, more practically, data chunks dictated by some specific memory or storage device format (e.g., a hard-drive sector formed by 512 bytes), that cannot be further divided because of the specific read/write scheme.
requested file $W_{d_l}$ of user $l \in L$, and let $\mu \triangleq (\mu_1, \ldots, \mu_L)$ denote the decoding functions of all the $L$ users. For a coded caching scheme defined by the triple $\mathfrak{F} \triangleq (\psi, \phi, \mu)$, the probability of error is defined as $P_e(\mathfrak{F}) \triangleq \max_{d \in \mathbb{N}^L} \max_{l \in L} \Pr \left[ \hat{W}_{d_l} \neq W_{d_l} \right]$. For a given finite file size $F$, a coded caching scheme $\mathfrak{F}$ is called admissible if $P_e(\mathfrak{F}) = 0$. Since the shared link is error free, such admissible schemes exist for all $F \in \mathbb{N}$.

Given an admissible coded caching scheme $\mathfrak{F}$ and the request vector $d$, let $R(M, L, \mathfrak{F}, d)$ be the length (expressed in data units) of the multicast message $Y$, where $R(M, L, \mathfrak{F}, d)$ represents the normalized load of the shared link expressed in file units. Let

$$R(M, L, \mathfrak{F}) \triangleq \max_{d \in \mathbb{N}^L} R(M, L, \mathfrak{F}, d)$$

denote the worst-case load of the shared link, for some $M \in [0, N]$ [12]. Note that when $\mathfrak{F}$ is the uncoded caching scheme, the worst-case load is given by $L(1 - \gamma)$ [12]. When $M = 0$, the trivially optimal scheme consists of transmitting the union of all requested files over the shared link in the delivery phase, resulting in $R(0, L, \mathfrak{F}) = L$ for $N \geq L$. When $M = N$, the obviously optimal scheme consists of caching the whole database at each user, resulting in $R(N, L, \mathfrak{F}) = 0$. When $M \in (0, N)$, we have $R(N, L, \mathfrak{F}) > 0$ for all admissible $\mathfrak{F}$. Then, let

$$G(M, L, \mathfrak{F}) \triangleq \frac{L(1 - \gamma)}{R(M, L, \mathfrak{F})}$$

denote the worst-case coded caching gain of a coded caching scheme $\mathfrak{F}$, for all $M \in (0, N)$ [13]. The worst-case coded caching gain represents the reduction factor of the worst-case load with respect to the case of uncoded caching. In general, for all $M \in (0, N)$, we wish to minimize the worst-case load $R(M, L, \mathfrak{F})$ i.e., maximize the worst-case coded caching gain $G(M, L, \mathfrak{F})$ over all admissible caching schemes $\mathfrak{F}$. The minimization is with respect to the caching functions $\phi$, the server encoding function $\psi$, and the decoding functions $\mu$.

In addition, an admissible coded caching scheme $\mathfrak{F}$ determines the subpacketization level (the number of packets per file) $\hat{F}(M, L, \mathfrak{F})$. Assuming that $\frac{F}{F(M, L, \mathfrak{F})} \in \mathbb{N}$, the (minimum) required file size equals to the subpacketization level in value. Thus, we also wish to analyze the tradeoff between the subpacketization level and the worst-case coded caching gain for an admissible caching scheme $\mathfrak{F}$. 
III. DECENTRALIZED CODED CACHING SCHEME

A. Motivating Example

In this part, we use Shanmugam et al.’s centralized coded caching scheme in [13] as a motivating example of the proposed decentralized random coded caching scheme. Shanmugam et al.’s centralized coded caching scheme is applied to the case where $L$ is divisible by some integer $K$, i.e., $\frac{L}{K} \in \mathbb{N}$, as illustrate in Fig. 1. Its key idea is to divide the $L$ users into $L/K$ groups of size $K$ with the $j$-th group consisting of users $(j-1)K+1, \ldots, jK$ and apply Maddah-Ali–Niesen’s centralized coded caching scheme [12] to each group separately. Such scheme is reviewed briefly here for completeness. For given $N, M, K$, assume $t \triangleq K\gamma$ to be an integer in the set $\{1, 2, \ldots, K-1\}$. In the placement phase, each file $W_n$ is partitioned into $\left(\begin{array}{c} K \\ t \end{array}\right)$ non-overlapping packets of $\frac{F}{t}$ data units, where each packet is indexed by a distinct subset of $K \triangleq \{1, \ldots, K\}$ of size $t$. The collection of packets for file $W_n$ is given by $\{W_n, T : T \subset K, |T| = t\}$. Then, construct $K$ cache contents, denoted by $C_1, \ldots, C_K$, where

$$C_k \triangleq \{W_n, T : n \in \mathcal{N}, k \in T, T \subset K, |T| = t\}, \quad k \in K.$$ (1)

In short, cache content $C_k$ contains all packets of all files whose associated user subset $T$ contains the index $k$. In the placement phase, for each group $j$, user $(j-1)K+k$ stores cache content $C_k$. In the delivery phase, for each group $j$, the server transmits one coded multicast message $\oplus_{k \in S} W_{d_k, S \setminus \{k\}}$, for each subset $S \subseteq \{(j-1)K+1, \ldots, jK\}$ of cardinality $|S| = t+1$, where $\oplus$ denotes bitwise XOR. By Theorem 1 in [12], we have that each user is able to decode the requested file based on the received coded multicast messages and its cache content.

When $L$ is indivisible by $K$, i.e., $\frac{L}{K} \notin \mathbb{N}$, each of the first $\lceil L/K \rceil$ groups contains $K$ users, and the last group contains only $L - \lceil L/K \rceil K$ users. To apply Maddah-Ali–Niesen’s centralized coded caching scheme [12] to the last group, we can construct $\lceil L/K \rceil K - L$ virtual users with arbitrary file requests in the last group. Note that when $L - \lceil L/K \rceil K$ is small, this approach may be inefficient. Thus, it is desirable to design efficient delivery procedure for the case where the number of users in a group is smaller than the number of cache contents $K$.

Maddah-Ali–Niesen’s centralized coded caching [12] and Shanmugam et al.’s centralized coded caching scheme [13] achieve different tradeoffs between the subpacketization level and the worst-case coded caching gain. In particular, Maddah-Ali–Niesen’s centralized coded caching scheme [12] requires partitioning each
worst-case coded caching gain

coded caching requires partitioning of each file into

\[ F(M, L) = 2^{LH(\gamma)} \]
packets and achieves worst-case coded caching gain \( G_{MC}(M, L, K) = \frac{L(1-\gamma)}{\gamma} = 1 + L\gamma \), where \( H(\gamma) = -\gamma \ln \gamma - (1 - \gamma) \ln (1 - \gamma) \) denotes the binary entropy function. Thus, \( F_{MC}(M, L) \) grows exponentially with \( L \) and \( G_{MC}(M, L, K) \) grows as \( \Theta(L) \). Shanmugam et al.’s centralized coded caching requires partitioning of each file into \( F_{SC}(M, K) = \binom{K}{L} \approx 2^{KH(\gamma)} \) packets and achieves worst-case coded caching gain \( G_{SC}(M, L, K) = \frac{L(1-\gamma)}{K(1+\gamma)} = 1 + K\gamma \). We see how \( F_{SC}(M, K) \) and \( G_{SC}(M, L, K) \) grow with \( L \), by letting \( K \) to be a function of \( L \).

- If we let \( K = \Theta(L) \), then \( F_{SC}(M, K) \) is exponential with \( L \) and \( G_{SC}(M, L, K) \) grows as \( \Theta(L) \). Thus, in the case of \( K = \Theta(L) \), the scalings of the subpacketization level and the worst-case coded caching gain with \( L \) for Maddah-Ali–Niesen’s [12] and Shanmugam et al.’s [13] centralized coded caching schemes are the same.

- If we let \( K = O(1) \), then \( F_{SC}(M, K) \) is a constant and also \( G_{SC}(M, L, K) \) is a constant. That is, neither \( F_{SC}(M, K) \) nor \( G_{SC}(M, L, K) \) grows with \( L \).

- In general, we can obtain a family of scaling laws by choosing how \( K \) increases with \( L \). For example, if we let \( K = \Theta(\log_2 L) \), then \( F_{SC}(M, K) \) grows linearly with \( L \) and \( G_{SC}(M, L, K) \) grows as \( \Theta(\log_2 L) \). In doing so, we can find an acceptable regime for Shanmugam et al.’s centralized coded caching scheme [13] such that both the subpacketization level and the worst-case coded caching gain are

(a) Content placement of Shanmugam et al.’s centralized coded caching scheme [13]. \( Z_1 = Z_6 = C_1, \) \( Z_2 = Z_7 = Z_8 = C_3, \) \( Z_3 = Z_8 = C_4, \) and \( Z_5 = Z_{10} = C_5. \)

(b) Random content placement of the proposed decentralized random coded caching scheme. \( Z_2 = C_1, \) \( Z_1 = Z_7 = Z_3 = Z_6 = C_3, \) \( Z_4 = Z_5 = Z_3 = C_4, \) and \( Z_5 = C_5. \)

Fig. 1: Examples. \( L = 10, K = 5. \)
acceptable.

As illustrated in Section I, all decentralized coded caching schemes based on Maddah-Ali–Niesen’s decentralized coded caching scheme [13], [14], [16], [19]–[22] have the finite file size problem, i.e., cannot achieve desirable worst-case coded caching gains when the file size is limited. It is not clear whether the idea of Shanmugam et al.’s centralized coded caching scheme can be applied to design more efficient decentralized coded caching schemes which can achieve desirable worst-case coded caching gains when the file size is small.

B. Proposed Decentralized Random Coded Caching Scheme

In this part, motivated by Shanmugam et al.’s centralized coded caching scheme reviewed in Section III-A, we propose a new decentralized random coded caching scheme. This scheme, in the regime of practical moderate file size $F$, is able to outperform the classical Maddah-Ali–Niesen’s decentralized scheme of [14], as well as the schemes with improved delivery (e.g., [15], [16], [23]) based on the same random decentralized placement of Maddah-Ali–Niesen’s decentralized scheme [14], as discussed in Section I.

For given $K \in \{2, 3, \ldots \}$, we consider values of cache size given by $M \in M_K \triangleq \{N/K, 2N/K, \ldots, (K-1)N/K\}$. Other values of $M \in (0, N)$ can be handled by memory sharing as in [12]. We define a cache content base parameterized by $K$ consisting of a collection of $K$ cache contents, i.e., $C \triangleq \{C_1, C_2, \ldots, C_K\}$, where $C_k$ is given by (1). That is, the subpacketization level of the proposed scheme, denoted by $\hat{F}_r(M, K) \triangleq \binom{K}{K^*}$, is the same as that of Shanmugam et al.’s centralized coded caching scheme.

In the placement phase, each user $l \in \mathcal{L}$ independently chooses a cache content at random with uniform probability and stores it in its cache. We define the occupancy number $X_k$ of cache content $C_k$ as the number of users storing $C_k$, and let $X_{\text{max}} \triangleq \max_{k \in \mathcal{K}} X_k$. Note that the occupancy numbers $X \triangleq (X_k)_{k \in \mathcal{K}}$ essentially reflect the content placement in the sense that, by symmetry, the worst-case load is the same for all placements with the same $X$. We now introduce a $K \times X_{\text{max}}$ matrix $D \triangleq (D_{k,j})_{k \in \mathcal{K}, j = 1, \ldots, X_{\text{max}}}$, referred to as the user information matrix, to describe the cache contents and requests of all the users under the random placement. Specifically, for the $k$-th row of this matrix, let $D_{k,j} \in \mathcal{N}$ denote the index of the file requested by the user in the $j$-th group who stores $C_k$, if $j \in \{1, 2, \ldots, X_k\}$; set $D_{k,j}$ to be 0, if $j \in \{X_k + 1, X_k + 2, \ldots, X_{\text{max}}\}$. Let $\hat{\mathcal{K}}_j \triangleq \{k \in \mathcal{K} : D_{k,j} \neq 0\}$ denote the index set of the cache contents stored at the users in the $j$-th group. Thus, $\hat{\mathcal{K}}_j \triangleq |\hat{\mathcal{K}}_j| \leq K$ also represents the number of users
in the \( j \)-th group. Note that \( \widehat{K}_j \) is non-increasing with \( j \) and \( \sum_{j=1}^{X_{\text{max}}} \widehat{K}_j = L \). The relation between \( \widehat{K}_j \), \( j \in \{1, 2, \ldots, X_{\text{max}}\} \) and \( X \) is as follows. Let \( X(1) \leq X(2) \leq \ldots X(K-1) \leq X(K) \) be the \( X_k \)'s arranged in increasing order, so that \( X(k) \) is the \( k \)-th smallest. Note that \( X(K) = X_{\text{max}} \). Set \( X(0) = 0 \). For all \( j \in \{1, 2, \ldots, X_{\text{max}}\} \), \( \widehat{K}_j = K - k + 1 \) with \( k \) satisfying \( X(k-1) < j \leq X(k) \).

Example 1 (User Information Matrix): As illustrated in Fig. 1 (b), consider \( L = 10 \) and \( K = 5 \). Suppose the cache contents of these users are as follows: \( Z_2 = C_1 \), \( Z_1 = Z_7 = Z_9 = C_2 \), \( Z_3 = Z_6 = C_3 \), \( Z_4 = Z_5 = Z_{10} = C_4 \), and \( Z_8 = C_5 \). Then, we have \( X_1 = 1 \), \( X_2 = 3 \), \( X_3 = 2 \), \( X_4 = 3 \), \( X_5 = 1 \), \( \widehat{K}_1 = \{1, 2, 3, 4, 5\} \), \( \widehat{K}_2 = \{2, 3, 4\} \), \( \widehat{K}_3 = \{2, 4\} \), and the user information matrix is

\[
D = (D_{k,j})_{k \in \widehat{K}, j=1,\ldots,X_{\text{max}}} = \begin{bmatrix}
d_2 & 0 & 0 \\
d_1 & d_7 & d_9 \\
d_3 & d_6 & 0 \\
d_4 & d_5 & d_{10} \\
d_8 & 0 & 0
\end{bmatrix}, \tag{2}
\]

\[\diamondsuit\]

In the delivery phase, the users in each group are served simultaneously using coded-multicasting. Since the users in each group have distinct cache contents, they can be served using the delivery procedure of Shanmugam et al.’s centralized scheme [13], reviewed in Section III-A. However, for all \( j \) such that \( \widehat{K}_j < K \) this would be inefficient, since \( K - \widehat{K}_j \) requests are effectively not there. In the following, we present a delivery procedure that takes explicitly into account that each group has generally less than \( K \) users. This is essential to achieve a small overall worst-case load.

Consider the \( j \)-th group. Denote \( \tau_j \triangleq \min\{t+1, \widehat{K}_j\} \) and \( \overline{\tau}_j \triangleq \max\{1, t+1 - (K - \widehat{K}_j)\} \). Consider any \( \tau_j \in \{\overline{\tau}_j, \overline{\tau}_j + 1, \ldots, \overline{\tau}_j\} \). We focus on a subset \( S_j^1 \subseteq \widehat{K}_j \) with \( |S_j^1| = \tau_j \) and a subset \( S_j^2 \subseteq \mathcal{K} - \widehat{K}_j \) with \( |S_j^2| = t + 1 - \tau_j \). Observe that every \( \tau_j - 1 \) cache contents in \( S_j^1 \) share a packet that is needed by the user which stores the remaining cache content in \( S_j^1 \). More precisely, for any \( s \in S_j^1 \), the packet \( W_{D_s,j}(S_j^1 \cup \{s\}) \cup S_j^2 \) is requested by the user storing cache content \( s \), since it is a packet of \( W_{D_s,j} \). At the same time, it is missing at cache content \( s \) since \( s \notin S_j^1 \setminus \{s\} \). Finally, it is present in the cache content \( k \in S_j^1 \setminus \{s\} \). For any

\[\text{By taking all values of } \tau_j \text{ in } \{\overline{\tau}_j, \overline{\tau}_j + 1, \ldots, \overline{\tau}_j\}, \text{ we can go through all subsets } S_j^1 \subseteq \widehat{K}_j \text{ and } S_j^2 \subseteq \mathcal{K} - \widehat{K}_j, \text{ such that } S_j^1 \neq \emptyset \text{ and } |S_j^1 \cup S_j^2| = t + 1.\]
subset $S_j^1$ of cardinality $|S_j^1| = \tau_j$ and subset $S_j^2$ of cardinality $|S_j^2| = t + 1 - \tau_j$, the server transmits coded multicast message $\oplus_{s \in S_j^1} W_{D_{s,j},(S_j^3 \setminus \{s\}) \cup S_j^2}$.

In Algorithm 1, we formally describe the delivery procedure for the users in the $j$-th group. Note that when $\tilde{K}_j = K$, the proposed delivery procedure for the $j$-th group in Algorithm 1 reduces to the one in Maddah-Ali–Niesen’s centralized scheme [12]. The delivery procedure for the $j$-th group is repeated for all groups $j = 1, \ldots, X_{\text{max}}$, and the multicast message $Y$ is simply the concatenation of the coded multicast messages for all groups $j = 1, \ldots, X_{\text{max}}$.

**Example 2 (Content Delivery):** Consider $L = 10$, $K = 5$ and the same content placement as in Example 1. Suppose $N = 5$ and $M = 2$. Thus, $t = K\gamma = 2$. According to Algorithm 1, the coded multicast messages for the first group are equivalent to those generated by the delivery procedure of Maddah-Ali–Niesen’s centralized scheme [12]. The coded multicast messages for the second group and the third group are illustrated in Table I and Table II, separately.

Now, we argue that each user can successfully recover its requested file. Consider the user in the $j$-th group that stores cache content $k$. Consider subsets $S_j^1 \subseteq \tilde{K}_j$ and $S_j^2 \subseteq K - \tilde{K}_j$, such that $k \in S_j^1$ and $|S_j^1 \cup S_j^2| = t + 1$. Since cache content $k \in S_j^1$ already contains the packets $W_{D_{s,j},(S_j^3 \setminus \{s\}) \cup S_j^2}$ for all $s \in S_j^1 \setminus \{k\}$, the user storing cache content $k$ can solve $W_{D_{k,j},(S_j^3 \setminus \{k\}) \cup S_j^2}$ from the coded multicast message $\oplus_{s \in S_j^1} W_{D_{s,j},(S_j^3 \setminus \{s\}) \cup S_j^2}$ sent over the shared link. Since this is true for every such subsets $S_j^1 \subseteq \tilde{K}_j$ and $S_j^2 \subseteq K - \tilde{K}_j$ satisfying $k \in S_j^1$ and $|S_j^1 \cup S_j^2| = t + 1$, the user in the $j$-th group storing cache content $k$ is able to recover all packets of the form $\left\{ W_{D_{k,j},(S_j^3 \setminus \{k\}) \cup S_j^2} : S_j^1 \subseteq \tilde{K}_j, S_j^2 \subseteq K - \tilde{K}_j, k \in S_j^1, |(S_j^1 \setminus \{k\}) \cup S_j^2| = t \right\} = \left\{ W_{D_{k,j},T} : T \subseteq K \setminus \{k\}, |T| = t \right\}$. The remaining packets are of the form $\left\{ W_{D_{k,j},T} : k \in T, T \subseteq K, |T| = t \right\}$.

But these packets are already contained in cache content $k$. Hence, the user in the $j$-th group storing cache content $k$ can recover all packets of its requested file $W_{D_{k,j}}$. The overall scheme is formally summarized.

**Algorithm 1** Delivery Algorithm for Group $j$

1. initialize $\tau_j \leftarrow \min\{t + 1, \tilde{K}_j\}$, $\tau_j \leftarrow \max\{1, t + 1 - (K - \tilde{K}_j)\}$ and $t \leftarrow \frac{KM}{N}$.
2. for $\tau_j = \tau_j^2 : \tau_j$ do
3. for all $S_j^1 \subseteq \tilde{K}_j, S_j^2 \subseteq K - \tilde{K}_j : |S_j^1| = \tau_j, |S_j^2| = t + 1 - \tau_j$ do
4. server sends $\oplus_{s \in S_j^1} W_{D_{s,j},(S_j^3 \setminus \{s\}) \cup S_j^2}$
5. end for
6. end for
TABLE I: Coded multicast message for the second group. $\tau_2 = 3$, $\tau_2 = 1$, $t = 2$, and $K_2 = \{2, 3, 4\}$.

| $\tau_2$ | $S_2^1$ | $S_2^2$ | Coded Multicast Message |
|----------|---------|---------|-------------------------|
| 3        | (2, 3, 4) | $\emptyset$ | $W_{D_2,2,2}(3, 4) \oplus W_{D_3,2,2}(2, 4) \oplus W_{D_4,2,2}(2, 3)$ |
| 2        | (2, 3)    | $\{1\}$   | $W_{D_2,2,2}(3, 3) \oplus W_{D_3,2,2}(2, 1)$ |
| 2        | (2, 3)    | $\{5\}$   | $W_{D_2,2,2}(3, 5) \oplus W_{D_3,2,2}(2, 5)$ |
| 2        | (2, 4)    | $\{1\}$   | $W_{D_2,2,2}(4, 1) \oplus W_{D_4,2,2}(2, 3)$ |
| 2        | (2, 4)    | $\{5\}$   | $W_{D_2,2,2}(4, 5) \oplus W_{D_4,2,2}(2, 5)$ |
| 2        | (3, 4)    | $\{1\}$   | $W_{D_3,2,2}(4, 1) \oplus W_{D_4,2,2}(3, 1)$ |
| 2        | (3, 4)    | $\{5\}$   | $W_{D_3,2,2}(4, 5) \oplus W_{D_4,2,2}(3, 5)$ |
| 1        | (2)       | $\{1, 5\}$ | $W_{D_2,2,2}(1, 5)$ |
| 1        | (3)       | $\{1, 5\}$ | $W_{D_3,2,2}(1, 5)$ |
| 1        | (4)       | $\{1, 5\}$ | $W_{D_4,2,2}(1, 5)$ |

TABLE II: Coded multicast message for the third group. $\bar{\tau}_3 = 2$, $\tau_3 = 1$, $t = 2$, and $K_3 = \{2, 4\}$.

| $\tau_3$ | $S_3^1$ | $S_3^2$ | Coded Multicast Message |
|----------|---------|---------|-------------------------|
| 2        | $\{2, 4\}$ | $\{1\}$ | $W_{D_2,3,3}(4, 1) \oplus W_{D_4,3,3}(2, 1)$ |
| 2        | $\{2, 4\}$ | $\{3\}$ | $W_{D_2,3,3}(4, 3) \oplus W_{D_4,3,3}(2, 3)$ |
| 2        | $\{2, 4\}$ | $\{5\}$ | $W_{D_2,3,3}(4, 5) \oplus W_{D_4,3,3}(2, 5)$ |
| 1        | $\{2\}$  | $\{1, 3\}$ | $W_{D_2,3,3}(1, 3)$ |
| 1        | $\{2\}$  | $\{1, 5\}$ | $W_{D_2,3,3}(1, 5)$ |
| 1        | $\{2\}$  | $\{3, 5\}$ | $W_{D_2,3,3}(3, 5)$ |
| 1        | $\{4\}$  | $\{1, 3\}$ | $W_{D_4,3,3}(1, 3)$ |
| 1        | $\{4\}$  | $\{1, 5\}$ | $W_{D_4,3,3}(1, 5)$ |
| 1        | $\{4\}$  | $\{3, 5\}$ | $W_{D_4,3,3}(3, 5)$ |

**Algorithm 2 Decentralized Random Coded Caching Scheme**

**Placement Procedure**

1: for $l \in \mathcal{L}$ do  
2: $Z_l \leftarrow C_k$, where $k$ is chosen uniformly at random from $\mathcal{K}$  
3: end for

**Delivery Procedure**

1: for $j = 1, \cdots, X_{\text{max}}$ do  
2: Run Algorithm 1 for the users in the $j$-th group  
3: end for

in Algorithm 2.

**Remark 1 (Comparison with Existing Schemes):** The placement procedures of the decentralized schemes in [13], [16]–[18], [20]–[23] are based on Maddah-Ali–Niesen’s random placement procedure [14], which generates packets of random lengths and causes large variance of the lengths of packets in each coded multicast message, leading to the “bit waste” effect in coded multicast XOR operations of the delivery procedure [15] and a drastic reduction of coded-multicasting opportunities. In contrast, in the random placement procedure of the proposed scheme, each user randomly and independently selects its cache content from a carefully designed cache content base, ensuring that the packets involved in each coded multicast XOR operation have the same length, hence fundamentally avoiding the “bit waste” effect and
increasing coded-multicasting opportunities in the finite file size regime. In addition, when the number of users in one group is smaller than the number of cache contents $K$, the proposed delivery procedure is more efficient than the delivery procedure of Maddah-Ali–Niesen’s centralized scheme, as the proposed delivery procedure avoids transmitting the redundant coded-multicast messages

$$\bigoplus_{s \in S_j} W_{D_{s,j}, S_j^2 \setminus \{s\}}, \quad S_j^2 \subseteq K - \hat{K}_j,$$

$|S_j^2| = t + 1$ for each group $j \in \{1, \ldots, X_{\text{max}}\}$ with $\hat{K}_j < K$.

\[\therefore\]

IV. WORST-CASE LOAD ANALYSIS

In this section, we first analyze the worst-case load of the proposed scheme for finite file size. Then, we analyze the asymptotic worst-case load of the proposed scheme when the file size is large.

A. Worst-case Load

1) Worst-case Load of Proposed Scheme: First, consider occupancy numbers $X$. Denote $r(M, K, \hat{K}_j)$ as the worst-case load for serving $\hat{K}_j$ users in the $j$-th group.

**Lemma 1 (Per-group Worst-case Load):** The per-group worst-case load for serving $\hat{K}_j$ users in group $j$ is given by

$$r(M, K, \hat{K}_j) = \begin{cases} \binom{K}{\hat{K}_j + 1}, & \hat{K}_j + 1 > K(1 - \gamma) \\ \binom{K}{\hat{K}_j} - \binom{K - \hat{K}_j}{\hat{K}_j + 1}, & \hat{K}_j + 1 \leq K(1 - \gamma). \end{cases} \quad (3)$$

**Proof:** Please refer to Appendix A. \[\blacksquare\]

When $\hat{K}_j + 1 > K(1 - \gamma)$, $r(M, K, \hat{K}_j)$ is equal to the worst-case load of Maddah-Ali–Niesen’s centralized scheme [12] for serving $K$ users. When $\hat{K}_j + 1 \leq K(1 - \gamma)$, the first term of $r(M, K, \hat{K}_j)$ is equal to the worst-case load of Maddah-Ali–Niesen’s centralized scheme for serving $K$ users, and the second term indicates the worst-case load reduction enabled by the proposed delivery procedure.

Based on Lemma 1, we now obtain the worst-case load for serving all $L$ users. Let $R(M, L, K, X)$ denote the worst-case load for serving all the users for given $X$. It is clear that $R(M, L, K, X) = \sum_{j=1}^{X_{\text{max}}} r(M, K, \hat{K}_j)$. Note that the joint distribution of random variables $\hat{K}_j$, $j \in \{1, 2, \ldots, X_{\text{max}}\}$ is hard to derive. In contrast, $X$ follows a multinomial distribution, which is known. Thus, to facilitate the calculation
of the average\textsuperscript{5} worst-case load over the random occupancy numbers $X$, we first express $R(M, L, K, X)$ as a function of $X$, as shown in the following lemma.

**Lemma 2 (Worst-case Load for All Users):** The worst-case load for serving all the users for given $X$ is given by

$$R(M, L, K, X) = \frac{1}{K^{(K\gamma)}} \sum_{k=K\gamma+1}^{K} X(k) \binom{k-1}{K\gamma}.$$  \hspace{1cm} (4)

**Proof:**

$$R(M, L, K, X) = \sum_{j=1}^{X_{\text{max}}} r(M, K, \hat{K}_j) \overset{(a)}{=} \sum_{k=1}^{K} (X(k) - X(k-1)) r(M, K, K - k + 1)$$

$$\overset{(b)}{=} \sum_{k=1}^{t+1} (X(k) - X(k-1)) \binom{K}{K\gamma} \frac{(t+1)}{K\gamma} + \sum_{k=t+2}^{K} (X(k) - X(k-1)) \binom{K}{K\gamma} \frac{(t+1) - (k-1)}{K\gamma},$$

$$\overset{(c)}{=} \frac{1}{K\gamma} \sum_{k=t+1}^{K} X(k) \binom{k-1}{t} \overset{(d)}{=} \frac{1}{K\gamma} \sum_{k=K\gamma+1}^{K} X(k) \binom{k-1}{K\gamma},$$

where (a) is due to the fact that for all $j \in \mathbb{N}$, $\hat{K}_j = K - k + 1$ with $k$ satisfying $X(k-1) < j \leq X(k)$, (b) is due to Lemma 1, (c) is due to Pascal’s identity, i.e., $\binom{k+1}{t} = \binom{k}{t} + \binom{k}{t-1}$, and (d) is due to $t = K\gamma$. \hfill \Box

Let $R_r(M, L, K)$ denote the (average) worst-case load, where the average is over the random occupancy numbers $X$. Based on Lemma 2 and the multinomial distribution of $X$, we have the following result.

**Theorem 1 (Worst-case Load):** For a cache content base of cardinality $K \in \{2, 3, \cdots\}$ and $L \in \mathbb{N}$ users each with cache size $M \in \mathcal{M}_K$,

$$R_r(M, L, K) = \sum_{(x_1, x_2, \ldots, x_K) \in \mathcal{X}_{K,L}} \left( \begin{array}{c} L \\ x_1 x_2 \ldots x_K \end{array} \right) \frac{1}{K^L} \times \frac{1}{K^{(K\gamma)}} \sum_{k=K\gamma+1}^{K} x(k) \binom{k-1}{K\gamma},$$  \hspace{1cm} (5)

where $\mathcal{X}_{K,L} \triangleq \left\{ (x_1, x_2, \ldots, x_K) \in \mathbb{N}^K : \sum_{k=1}^{K} x_k = L \right\}$ for all $K \in \{2, 3, \cdots\}$ and $L \in \mathbb{N}$. \hfill \Box

Theorem 1 shows the worst-case load of the proposed scheme for finite $K$.

\textsuperscript{5}In this paper, for simplicity, we also refer to the average worst-case load of a decentralized coded caching scheme with random placement as the worst-case load.
2) \textbf{Worst-case Load Comparison with Maddah-Ali–Niesen’s and Shanmugam et al.’s Decentralized Schemes:}

First, we compare the worst-case load of the proposed scheme with Maddah-Ali–Niesen’s decentralized scheme [14]. Let $\hat{F}_{MD}(M, L)$ denote the subpacketization level of Maddah-Ali–Niesen’s decentralized scheme. Let $R_{MD}(M, L, \hat{F}_{MD}(M, L))$ denote the worst-case load under Maddah-Ali–Niesen’s decentralized scheme. By comparing the worst-case load of the proposed scheme with the lower bound on the worst-case load of Maddah-Ali–Niesen’s decentralized coded caching scheme given by Theorem 5 of [13], we have the following result.

\textbf{Theorem 2 (Load Comparison with Maddah-Ali–Niesen’s Decentralized Scheme):} For a cache content base of cardinality $K \in \{2, 3, \cdots\}$ and cache size $M \in \mathcal{M}_K$, there exists $L_r(M, K) > 0$, such that when $L > L_r(M, K)$, $R_{MD}(M, L, \hat{F}_{MD}(M, L)) > R_r(M, L, K)$, where $\hat{F}_{MD}(M, L) = \hat{F}(M, K)$.

\textit{Proof:} Please refer to Appendix B.

Theorem 2 indicates that, when the number of users is above a threshold, given the same subpacketization level, the worst-case load of the proposed scheme is smaller than that of Maddah-Ali–Niesen’s decentralized scheme. This demonstrates that the proposed scheme outperforms Maddah-Ali–Niesen’s decentralized scheme in the finite file size regime, when the number of users is sufficiently large.

Next, we compare the worst-case load of the proposed scheme with Shanmugam et al.’s decentralized coded caching scheme [13]. Let $\hat{F}_{SD}(M, L, g)$ denote the subpacketization level under Shanmugam et al.’s decentralized scheme, and let $R_{SD}(M, L, \hat{F}_{SD}(M, L, g), g)$ denote the worst-case load under Shanmugam et al.’s decentralized scheme, where the system parameter $g \in \mathbb{N}$ satisfies $\frac{L}{\left\lceil \frac{1}{\gamma} \right\rceil 3g \ln \left( \frac{1}{\gamma} \right)} \in \mathbb{N}$ [13]. For purpose of comparison, we need a lower bound on the worst-case load and a lower bound on the required file size of Shanmugam et al.’s decentralized scheme, which are given by the following lemma.

\textbf{Lemma 3 (Lower Bounds on Load and Subpacketization Level of Shanmugam et al.’s Decentralized Scheme):} For $L \in \mathbb{N}$ users each with cache size $M \in (0, N)$, when $\gamma \leq \frac{1}{8}$,

$$R_{SD}(M, L, \hat{F}_{SD}(M, L, g), g) \geq \frac{L}{g + 1} c \left( M, \hat{F}_{SD}(M, L, g), g \right),$$

(6)
and

$$
\hat{F}_{SD}(M, L, g) > \left(1 - \frac{R_{SD}(M, L, \hat{F}_{SD}(M, L, g), g)}{L(1 - \gamma)(1 - g)} \left(1 - \frac{\hat{F}_{SD}(M, L, g, g)}{\hat{F}_{SD}(M, L, g)}\right)\right) \frac{\left(K'\right)}{(g + 1)} \left(1 - \frac{\hat{F}_{SD}(M, L, g, g)}{\hat{F}_{SD}(M, L, g)}\right),
$$

(7)

where $K' \triangleq \left\lceil \left\lceil \frac{1}{\gamma} \right\rceil 3g \ln \left(\frac{1}{\gamma}\right) \right\rceil$, $d(M, g) \triangleq \frac{3g\left\lceil \frac{1}{\gamma} \right\rceil \ln \left(\frac{1}{\gamma}\right)}{\gamma}$, $\delta \triangleq 1 - \frac{1}{3d(M, g)}$, $c(M, \hat{F}_{SD}(M, L, g, g)) \triangleq (1 - g)(1 - \gamma)$, and $\theta(M, g) \triangleq \left(\frac{e^{-\delta}}{(1-\delta)^{1-\delta}}\right)^{K'\frac{1}{\gamma}} K'\frac{1}{\gamma}$.

Proof: Please refer to Appendix C.

By comparing the subpacketization level of the proposed scheme with the lower bound on the subpacketization level of Shanmugam et al.'s decentralized scheme given by (7) and using (6), we have the following result.

Theorem 3 (Worst-case Load Comparison with Shanmugam et al.'s Decentralized Scheme): For a cache content base of cardinality $K \in \{2, 3, \ldots\}$ and cache size $M \in \mathcal{M}_K$, there exist $q_r > 0$ and $\bar{L}_r(M, K) > 0$, such that for $\frac{1}{\gamma} > q_r$ and $L > \bar{L}_r(M, K)$, if $R_{SD}(M, L, \hat{F}_{SD}(M, L, g), g) = R_r(M, L, K)$ then $\hat{F}_{SD}(M, L, g) > \bar{F}_r(M, K)$.

Proof: Please refer to Appendix D.

Theorem 3 indicates that, when the number of users is above a threshold and the normalized local cache size is below a threshold, to achieve the same worst-case load, the subpacketization level of Shanmugam et al.'s decentralized scheme is larger than that of the proposed scheme. This demonstrates that the proposed scheme outperforms Shanmugam et al.'s decentralized scheme in the finite file size regime, when the number of users is sufficiently large and the normalized local cache size is sufficiently small.

3) Numerical Results: Fig. 2 illustrates the worst-case loads of the proposed scheme, Maddah-Ali–Niesen's centralized and decentralized coded caching schemes as well as Shanmugam et al.'s decentralized coded caching scheme versus $K$ when $N = 60$ and $M = 20$. For the proposed scheme, Maddah-Ali–Niesen's decentralized scheme and Shanmugam et al.'s decentralized scheme, each file is split into $\binom{K}{K/\gamma}$ nonoverlapping packets of equal size, while for Maddah-Ali–Niesen's centralized scheme, each file is split into $\binom{L}{L/\gamma}$ nonoverlapping packets of equal size. In the following, we discuss the observations made from Fig. 2.
Fig. 2: Worst-case load versus $K$ at $N = 60$ and $M = 20$. For Shanmugam et al.’s decentralized scheme [13], we choose $g = 2$. Note that when $N = 60$, $M = 20$ and $g = 2$, Shanmugam et al.’s decentralized scheme can be applied only to the case where $\frac{L}{20} \in \mathbb{N}$. Thus, Shanmugam et al.’s decentralized scheme is not shown in Fig. 2 (a). The red curve indicates the worst-case load of Maddah-Ali–Niesen’s decentralized scheme [14] when the file size goes to infinity.

First, we compare the worst-case loads of these coded caching schemes.

- **Maddah-Ali–Niesen’s centralized coded caching scheme:** Maddah-Ali–Niesen’s centralized scheme [12] achieves the minimum worst-case load among all the schemes. This is because assuming the number of users $L$ in the delivery phase is known in the placement phase, the centralized scheme carefully designs the content placement to maximize coded-multicasting opportunities among all users in the delivery phase.

- **Proposed decentralized random coded caching scheme:** When $L$ is moderate or large, the proposed scheme achieves a smaller worst-case load than Maddah-Ali–Niesen’s [14] and Shanmugam et al.’s [13] decentralized schemes, which verifies Theorem 2 and Theorem 3. In addition, when $K$ is small, the proposed scheme achieves a smaller worst-case load than Maddah-Ali–Niesen’s decentralized scheme. This is because in these two regimes, the random placement procedures in Maddah-Ali–Niesen’s and Shanmugam et al.’s decentralized schemes yield large variance of the lengths of packets involved in each coded multicast XOR operation, leading to a drastic reduction of coded-multicasting opportunities.

- **Maddah-Ali–Niesen’s and Shanmugam et al.’s decentralized coded caching schemes:** When $K$ is small, Shanmugam et al.’s decentralized scheme [13] achieves a larger worst-case load than Maddah-
Ali–Niesen’s decentralized scheme [14]. This is because the drawback of the “pull down phase” in Shanmugam et al.’s decentralized scheme (i.e., sacrificing coded-multicasting opportunities in coding packets of higher types) is significant when $K$ is small. When $K$ is large, Shanmugam et al.’s decentralized scheme achieves smaller worst-case load than Maddah-Ali–Niesen’s decentralized scheme. This is because the advantage of “pull down phase” in Shanmugam et al.’s decentralized scheme (i.e., increasing coded-multicasting opportunities in coding packets of lower types via alleviating the “bit waste” effect) is significant when $K$ is large.

Next, we explain the decrease of the worst-case load change with $K$ for each decentralized coded caching scheme.

- **Decentralized random coded caching scheme:** When $K$ increases, more users may lie in one group, and hence more users can make use of coded-multicasting opportunities. When $K$ further increases after reaching $L$, the waste of coded-multicasting opportunities increases due to lack of users. Overall, when $K$ increases, coded-multicasting opportunities among all users increase, and hence the worst-case load decreases.

- **Maddah-Ali–Niesen’s and Shanmugam et al.’s decentralized coded caching schemes:** When $K$ increases, the variance of the lengths of packets involved in each coded multicast XOR operation decreases, and hence coded-multicasting opportunities among all users increase. Thus, when $K$ increases, the worst-case loads of the two schemes decrease.

### B. Asymptotic Worst-case Load

Let

$$R_\infty(M, L) \triangleq \left(\frac{1}{\gamma} - 1\right) \left(1 - (1 - \gamma)^L\right)$$

denote the limiting worst-case load of Maddah-Ali–Niesen’s decentralized scheme. In the following, we study the asymptotic load of the proposed scheme.

**Lemma 4 (Asymptotic Worst-case Load):** For $L \in \mathbb{N}$ users each with cache size $M \in (0, N)$, $\Pr[\hat{K}_1 = L] \to 1$ as $K \to \infty$, and

$$R_{r,\infty}(M, L) \triangleq \lim_{K \to \infty} R_r(M, L, K) = R_\infty(M, L),$$

(9)
Fig. 3: Worst-case load versus $K$ at $L = 4$, $N = 4$ and $M = 2$. Expressions $R_{\infty}(M, L) + \frac{A(M, L)}{K}$ indicates the dominant term of the upper bound on $R_r(M, L, K)$.

where $R_{\infty}(M, L)$ is given by (8). Furthermore, for $L \in \{2, 3, \cdots \}$ users each with cache size $M \in (0, N)$,

$$R_r(M, L, K) \leq R_{\infty}(M, L) + \frac{A(M, L)}{K} + o\left(\frac{1}{K}\right), \text{ as } K \to \infty,$$

where

$$A(M, L) \triangleq \frac{1}{\gamma} \left(\frac{1}{\gamma} - 1\right) \left((1 - \gamma)^L - 1 + \frac{(L + 2)(L - 1)M}{2N}\right) - 1 + \frac{L(L - 1)M}{2N} \left(\frac{LM}{N} - 1\right) \geq 0.$$

(11)

Proof: Please refer to Appendix E.

Lemma 4 shows that as $K \to \infty$, the worst-case load of the proposed scheme converges to the same limiting worst-case load as that of Maddah-Ali–Niesen’s decentralized scheme [14], i.e., $R_{r,\infty}(M, L) = R_{\infty}(M, L)$. By Corollary 2 of [25], we know that no decentralized scheme can further reduce the worst-case load when $K \to \infty$ and $N \geq L$. In other words, Lemma 4 implies that the proposed scheme attains an exact order-optimal memory-load tradeoff when $K \to \infty$ and $N \geq L$. Furthermore, Lemma 4 indicates that the upper bound on $R_r(M, L, K)$ decreases with $K$ for large $K$ (due to $A(M, L) \geq 0$), and $R_r(M, L, K) = R_{\infty}(M, L) + O\left(\frac{1}{K}\right)$ as $K \to \infty$. Fig. 3 verifies Lemma 4.
V. WORST-CASE CODED CACHING GAIN ANALYSIS

In this section, we first analyze the worst-case coded caching gain of the proposed scheme, defined in Section II, as a function of the subpacketization level, in the regime of practical moderate file size. Then, we analyze the growth of the worst-case coded caching gain with respect to the subpacketization level, in the regime of large file size.

A. Worst-case Coded Caching Gain

1) Worst-case Coded Caching Gain of Proposed Scheme: In the following, we study the worst-case coded caching gain of the proposed scheme, defined as

\[ G_r(M, L, K) \triangleq \frac{L(1 - \gamma)}{R_r(M, L, K)}. \]

For finite \( K \), the relationship between \( G_r(M, L, K) \) and \( \hat{G}_r(M, K) \) is summarized in the following theorem.

**Theorem 4 (Worst-case Coded Caching Gain of Proposed Scheme):**

(i) For a cache content base of cardinality \( K \in \{2, 3, \cdots\} \) and \( L \in \{2, 3, \cdots\} \) users each with cache size \( M \in M_K \), \( 1 \leq G_r(M, L, K) < 1 + K\gamma \), and \( \lim_{L \to \infty} G_r(M, L, K) = 1 + K\gamma \). (ii) For a cache content base of cardinality \( K \in \{2, 3, \cdots\} \) and \( L \in \left\{ \left[ \frac{1}{2} \left( \frac{1}{\gamma} \right)^2 \right], \left[ \frac{1}{2} \left( \frac{1}{\gamma} \right)^2 \right] + 1, \cdots \right\} \) users each with cache size \( M \in M_K \), \( \left( \frac{1}{\gamma} \right)^{G_r(M, L, K) - 1} < \hat{G}_r(M, K) \leq \left( \frac{1}{\gamma} \right)^{\sqrt{2L\gamma - G_r(M, L, K) \tfrac{1}{\gamma}}} \), when \( G_r(M, L, K) \in \left[ 1, \min \left\{ \sqrt{2L\gamma}, 1 + K\gamma \right\} \right) \).

**Proof:** Please refer to Appendix G.

Recall that \( G_{SC}(M, L, K) = 1 + K\gamma \). Thus, \( G_{SC}(M, L, K) = \lim_{L \to \infty} G_r(M, L, K) \). In addition, recall that the proposed scheme and Shanmugam et al.’s decentralized scheme [13] have the same subpacketization level. Therefore, Theorem 4 indicates that, when \( L \to \infty \), the proposed decentralized coded caching scheme achieves the same tradeoff between the subpacketization level and the worst-case coded caching gain as Shanmugam et al.’s centralized coded caching scheme [13] for all \( K \).

Note that \( \hat{G}_r(M, K) \) increases with \( K \). In addition, from Fig. 2, we can observe that \( R_r(M, L, K) \) decreases with \( K \). Thus, we know that \( \hat{G}_r(M, K) \) increases with \( G_r(M, L, K) \). We can easily verify that the lower bound and the upper bound on \( \hat{G}_r(M, K) \) given in Theorem 4 also increase with \( G_r(M, L, K) \), when \( L \in \left\{ \left[ \frac{1}{2} \left( \frac{1}{\gamma} \right)^2 \right], \left[ \frac{1}{2} \left( \frac{1}{\gamma} \right)^2 \right] + 1, \cdots \right\} \). Fig. 4 illustrates Theorem 4.

2) Worst-case Coded Caching Gain Comparison with Maddah-Ali–Niesen’s and Shanmugam et al.’s Decentralized Schemes: First, we compare the worst-case coded caching gain of the proposed scheme with Maddah-Ali–Niesen’s decentralized scheme [14]. Theorem 5 of [13] shows that to achieve a constant worst-
Fig. 4: The subpacketization level versus the worst-case coded caching gain at $N = 4$, $M = 2$, $L = 48$ and $K = 2, 4, 6, 8$.

case coded caching gain larger than 2, the subpacketization level under Maddah-Ali–Niesen’s decentralized scheme is $\Omega\left(\frac{1}{L}e^{2L\gamma(1-\gamma)}\right)$ as $L \to \infty$, and hence the required file size goes to infinity when $L \to \infty$. In contrast, Theorem 4 indicates that, to achieve the same constant worst-case coded caching gain, the subpacketization level is $O(1)$ when $L \to \infty$. Therefore, to achieve the same constant worst-case coded caching gain, the required file size of the proposed scheme is much smaller than that of Maddah-Ali–Niesen’s decentralized scheme, when the number of users is large.

Next, we compare the worst-case coded caching gain of the proposed scheme with Shanmugam et al.’s decentralized coded caching scheme [13]. By Theorem 3, we know that to achieve the same worst-case coded caching gain, the subpacketization level of the proposed scheme is smaller than that of Shanmugam et al.’s decentralized scheme, when the number of users is large and the normalized local cache size is small.

B. Asymptotic Worst-case Coded Caching Gain

Let $G_{\infty}(M, L) \triangleq \frac{L(1-\gamma)}{R_{\infty}(M, L)} = \frac{L\gamma}{1-(1-\gamma)L}$ denote the limiting worst-case coded caching gain of Maddah-Ali–Niesen’s decentralized scheme. In the following, we study the asymptotic worst-case coded caching gain of the proposed scheme.

Lemma 5 (Asymptotic Worst-case Coded Caching Gain): For $L \in \mathbb{N}$ users each with cache size $M \in$
Worst-case Coded Caching Gain

\[ G_r(M, L, K) \]

\[ G_\infty(M, L) \]

\[ \left( 1 - \frac{A(M, L)H(\gamma)}{R_\infty(M, L) \ln \hat{F}_r(M, K)} \right) \]

\[ \ln \hat{F}_r(M, K) \]

\[ G_\infty(M, L) \]

\[ G_r(M, L) \]

\[ G_\infty(M, L) \]

\[ G_r(M, L, K) \]

\[ G_\infty(M, L) \]

\[ G_r(M, L) \]

\[ G_\infty(M, L) \]

\[ G_r(M, L, K) \]

\[ G_\infty(M, L) \]

\[ G_r(M, L, K) \]

\[ G_\infty(M, L) \]

\[ G_r(M, L, K) \]

\[ G_\infty(M, L) \]

\[ G_r(M, L, K) \]

\[ G_\infty(M, L) \]

\[ G_r(M, L, K) \]

\[ G_\infty(M, L) \]

\[ G_r(M, L, K) \]

\[ G_\infty(M, L) \]

\[ G_r(M, L, K) \]

\[ G_\infty(M, L) \]

\[ G_r(M, L, K) \]

\[ G_\infty(M, L) \]

\[ G_r(M, L, K) \]

\[ G_\infty(M, L) \]

\[ G_r(M, L, K) \]

\[ G_\infty(M, L) \]

\[ G_r(M, L, K) \]

\[ G_\infty(M, L) \]

\[ G_r(M, L, K) \]

\[ G_\infty(M, L) \]

\[ G_r(M, L, K) \]

\[ G_\infty(M, L) \]

\[ G_r(M, L, K) \]

\[ G_\infty(M, L) \]

\[ G_r(M, L, K) \]

\[ G_\infty(M, L) \]

\[ G_r(M, L, K) \]

\[ G_\infty(M, L) \]

\[ G_r(M, L, K) \]

\[ G_\infty(M, L) \]

\[ G_r(M, L, K) \]

\[ G_\infty(M, L) \]

\[ G_r(M, L, K) \]

\[ G_\infty(M, L) \]

\[ G_r(M, L, K) \]

\[ G_\infty(M, L) \]

\[ G_r(M, L, K) \]

\[ G_\infty(M, L) \]

\[ G_r(M, L, K) \]

\[ G_\infty(M, L) \]

\[ G_r(M, L, K) \]

\[ G_\infty(M, L) \]

\[ G_r(M, L, K) \]

\[ G_\infty(M, L) \]

\[ G_r(M, L, K) \]

\[ G_\infty(M, L) \]

\[ G_r(M, L, K) \]

\[ G_\infty(M, L) \]

\[ G_r(M, L, K) \]

\[ G_\infty(M, L) \]

\[ G_r(M, L, K) \]

\[ G_\infty(M, L) \]

\[ G_r(M, L, K) \]

\[ G_\infty(M, L) \]

\[ G_r(M, L, K) \]

\[ G_\infty(M, L) \]

\[ G_r(M, L, K) \]

\[ G_\infty(M, L) \]

\[ G_r(M, L, K) \]

\[ G_\infty(M, L) \]

\[ G_r(M, L, K) \]

\[ G_\infty(M, L) \]

\[ G_r(M, L, K) \]

\[ G_\infty(M, L) \]

\[ G_r(M, L, K) \]

\[ G_\infty(M, L) \]

\[ G_r(M, L, K) \]

\[ G_\infty(M, L) \]

\[ G_r(M, L, K) \]

\[ G_\infty(M, L) \]

\[ G_r(M, L, K) \]

\[ G_\infty(M, L) \]

\[ G_r(M, L, K) \]

\[ G_\infty(M, L) \]

\[ G_r(M, L, K) \]

\[ G_\infty(M, L) \]

\[ G_r(M, L, K) \]

\[ G_\infty(M, L) \]

\[ G_r(M, L, K) \]

\[ G_\infty(M, L) \]

\[ G_r(M, L, K) \]

\[ G_\infty(M, L) \]

\[ G_r(M, L, K) \]

\[ G_\infty(M, L) \]

\[ G_r(M, L, K) \]

\[ G_\infty(M, L) \]

\[ G_r(M, L, K) \]

\[ G_\infty(M, L) \]

\[ G_r(M, L, K) \]

\[ G_\infty(M, L) \]

\[ G_r(M, L, K) \]

\[ G_\infty(M, L) \]

\[ G_r(M, L, K) \]

\[ G_\infty(M, L) \]

\[ G_r(M, L, K) \]

\[ G_\infty(M, L) \]

\[ G_r(M, L, K) \]

\[ G_\infty(M, L) \]

\[ G_r(M, L, K) \]

\[ G_\infty(M, L) \]

\[ G_r(M, L, K) \]

\[ G_\infty(M, L) \]

\[ G_r(M, L, K) \]

\[ G_\infty(M, L) \]

\[ G_r(M, L, K) \]

\[ G_\infty(M, L) \]

\[ G_r(M, L, K) \]

\[ G_\infty(M, L) \]

\[ G_r(M, L, K) \]

\[ G_\infty(M, L) \]

\[ G_r(M, L, K) \]

\[ G_\infty(M, L) \]

\[ G_r(M, L, K) \]

\[ G_\infty(M, L) \]

\[ G_r(M, L, K) \]

\[ G_\infty(M, L) \]

\[ G_r(M, L, K) \]

\[ G_\infty(M, L) \]

\[ G_r(M, L, K) \]

\[ G_\infty(M, L) \]

\[ G_r(M, L, K) \]

\[ G_\infty(M, L) \]

\[ G_r(M, L, K) \]

\[ G_\infty(M, L) \]

\[ G_r(M, L, K) \]

\[ G_\infty(M, L) \]

\[ G_r(M, L, K) \]

\[ G_\infty(M, L) \]

\[ G_r(M, L, K) \]

\[ G_\infty(M, L) \]

\[ G_r(M, L, K) \]

\[ G_\infty(M, L) \]

\[ G_r(M, L, K) \]

\[ G_\infty(M, L) \]

\[ G_r(M, L, K) \]

\[ G_\infty(M, L) \]

\[ G_r(M, L, K) \]

\[ G_\infty(M, L) \]

\[ G_r(M, L, K) \]

\[ G_\infty(M, L) \]

\[ G_r(M, L, K) \]

\[ G_\infty(M, L) \]

\[ G_r(M, L, K) \]

\[ G_\infty(M, L) \]

\[ G_r(M, L, K) \]

\[ G_\infty(M, L) \]

\[ G_r(M, L, K) \]
is sufficiently large. Then, we showed that the proposed scheme achieves an order-optimal memory-load tradeoff. Next, we analyzed the worst-case coded caching gain of the proposed scheme, and characterized the corresponding subpacketization level. We also analyzed the growth of the worst-case coded caching gain of the proposed scheme with respect to the subpacketization level when the file size is large. Numerical results showed that the proposed scheme outperforms Maddah-Ali–Niesen’s and Shanmugam et al.’s decentralized schemes when the file size is limited.

APPENDIX A: PROOF OF LEMMA 1

First, we derive the expression of the total number of coded multicast messages sent by the server for serving the $\hat{K}_j$ users in the $j$-th group. Consider any $\tau_j$ satisfying $\tau_j \leq \tau_j \leq \tau_j$. For any subsets $S_j^1$ and $S_j^2$ of cardinalities $|S_j^1| = \tau_j$ and $|S_j^2| = t + 1 - \tau_j$, the server sends one coded multicast message, i.e., $\oplus_{s \in S_j^1} W_{D_{s,j},(S_j^1 \setminus \{s\}) \cup S_j^2}$, which is of $\frac{F}{(\tau_j)}$ data units. Since the number of such $S_j^1$ is $(\hat{K}_j)$ and the number of such $S_j^2$ is $(\frac{K - \hat{K}_j}{t + 1 - \tau_j})$, for given $\tau_j$, the number of coded multicast message sent by the server for serving the $\hat{K}_j$ users in the $j$-th group is $(\hat{K}_j) \cdot (\frac{K - \hat{K}_j}{t + 1 - \tau_j})$. Summing over all $\tau_j$, we can obtain the total number of coded multicast messages sent by the server for serving the $\hat{K}_j$ users in the $j$-th group, i.e., $\sum_{\tau_j=\tau_j}^{\tau_j} (\hat{K}_j) \cdot (\frac{K - \hat{K}_j}{t + 1 - \tau_j})$. Note that this holds for all $d \in N^L$. Next, we calculate $\sum_{\tau_j=\tau_j}^{\tau_j} (\hat{K}_j) \cdot (\frac{K - \hat{K}_j}{t + 1 - \tau_j})$ by considering two cases.

(i) When $\hat{K}_j + 1 \leq K(1 - \gamma)$, by using Vandermonde identity and Chu-Vandermonde identity, we have $\sum_{\tau_j=\tau_j}^{\tau_j} (\hat{K}_j) \cdot (\frac{K - \hat{K}_j}{t + 1 - \tau_j}) = (\frac{K}{(t + 1)}) - (\frac{K - \hat{K}_j}{t + 1})$. Thus, in this case, the total number of data units sent over the shared link for serving the $\hat{K}_j$ users in the $j$-th group is $\frac{F}{(\tau_j)} \left(\left(\frac{K}{(t + 1)}\right) - (\frac{K - \hat{K}_j}{t + 1})\right) = \frac{F}{(\tau_j)} \left(\frac{K}{(K + 1)} - (\frac{K - \hat{K}_j}{t + 1})\right)$.

(ii) When $\hat{K}_j + 1 > K(1 - \gamma)$, by using Vandermonde identity and Chu-Vandermonde identity, we have $\sum_{\tau_j=\tau_j}^{\tau_j} (\hat{K}_j) \cdot (\frac{K - \hat{K}_j}{t + 1 - \tau_j}) = (\frac{K}{(t + 1)})$. Thus, in this case, the total number of data units sent over the shared link for serving the $\hat{K}_j$ users in the $j$-th group is $\frac{F}{(\tau_j)} \left(\frac{K}{(t + 1)}\right) = \frac{F}{(K + 1)} \frac{K}{(K + 1)}$. Therefore, we can obtain $r(M, K, \hat{K}_j)$ in (3) and complete the proof of Lemma 1.

APPENDIX B: PROOF OF THEOREM 2

First, we calculate an upper bound on the worst-case load under the proposed scheme. Recall that under the proposed scheme, $X$ follows multinomial distribution. Thus, we have $\mathbb{E}[X_k] = \frac{L}{K}$ and $\text{Var}[X_k] =$
\[
L \frac{1}{K} (1 - \frac{1}{K}) \text{ for any } k \in \mathcal{K}. \text{ By [26, Proposition 2], we have}
\]
\[
\mathbb{E}_X[X_{(k)}] \leq \frac{L}{K} + \sqrt{\frac{L}{K} (1 - \frac{1}{K})} \sqrt{\frac{K}{2(K - k + 1)}}.
\] (14)

By (4), we have
\[
R_r(M, L, K) = \mathbb{E}_X \left[ \frac{1}{(K/K_\gamma)_{k=K_\gamma}} \sum_{k=K_\gamma+1}^{K} X_{(k)} \left( \frac{k-1}{K_\gamma} \right) \right] = \frac{1}{(K/K_\gamma)_{k=K_\gamma}} \sum_{k=K_\gamma+1}^{K} \left( \frac{k-1}{K_\gamma} \right) \mathbb{E}_X[X_{(k)}]
\]
\[
\leq \frac{L}{K} \frac{1}{(K/K_\gamma)_{k=K_\gamma+1}} \sum_{k=K_\gamma+1}^{K} \left( \frac{k-1}{K_\gamma} \right) + \sqrt{\frac{L}{K} (1 - \frac{1}{K})} \sqrt{\frac{K}{2}} \sum_{k=K_\gamma+1}^{K} \left( \frac{k-1}{K_\gamma} \right) \sqrt{\frac{K}{2(K - k + 1)}}
\]
\[
\leq \frac{L}{K} \frac{1}{(K/K_\gamma)_{k=K_\gamma+1}} \sum_{k=K_\gamma+1}^{K} \left( \frac{k-1}{K_\gamma} \right) + \frac{\sqrt{L}}{K_\gamma} \frac{1}{K_\gamma} \frac{K}{2} \sum_{k=K_\gamma+1}^{K} \left( \frac{k-1}{K_\gamma} \right).
\] (15)

where (a) is due to (14), (b) is due to \( \sqrt{\frac{K}{2(K - k + 1)}} \leq \sqrt{\frac{K}{2}} \) for all \( k \in \{ K_\gamma + 1, K_\gamma + 2, \ldots, K \} \) and (c) is due to Pascal’s identity, i.e., \( \binom{k+1}{t} = \binom{k}{t} + \binom{k}{t-1} \). Therefore, by (15), we have \( R_r(M, L, K) \leq R_{r_{ub}}(M, L, K) \), where \( R_{r_{ub}}(M, L, K) \triangleq \frac{(L + K \sqrt{\frac{L}{2}})(1 - \gamma)}{1 + K_\gamma} \).

Next, we obtain a lower bound on the worst-case load under Maddah-Ali–Niesen’s decentralized scheme. By [13, Theorem 5], we have \( R_{MD}(M, L, \hat{F}_{MD}(M, L)) \geq R_{m}(M, L, \hat{F}_{MD}(M, L)) \), where
\[
R_{m}(M, L, \hat{F}_{MD}(M, L)) \triangleq L(1 - \gamma) - \hat{F}_{MD}(M, L)L^2 \gamma e^{-2L \gamma (1 - \gamma)(1 - \frac{1}{K})}.
\] (16)

Substituting \( \hat{F}_{MD}(M, L) = \hat{F}_r(M, K) \) into (16), we have \( R_{m}(M, L, \hat{F}_{MD}(M, L)) = L(1 - \gamma) - \frac{(K)^2 \gamma^2}{e^{2L \gamma (1 - \gamma)(1 - \frac{1}{K})}} \).

Finally, we show that when \( L \) is above a threshold, \( R_{MD}(M, L, \hat{F}_{MD}(M, L)) > R_r(M, L, K) \) by showing \( R_{m}(M, L, \hat{F}_{MD}(M, L)) > R_{r_{ub}}(M, L, K) \), where \( \hat{F}_{MD}(M, L) = \hat{F}_r(M, K) \). Denote \( \varphi(L) \triangleq 1 - \gamma - \frac{(K) \gamma^2 e^{2L \gamma (1 - \gamma)(1 - \frac{1}{K})}}{1 + K_\gamma} \). Note that \( R_{m}(M, L, \hat{F}_r(M, K)) \) as \( L \rightarrow \infty \), we have \( \lim_{L \rightarrow \infty} \varphi(L) > 0 \). Thus, we know that there exists \( \mathcal{L}_r(M, K) > 0 \), such that when \( L > \mathcal{L}_r(M, K) \), \( \varphi(L) > 0 \), and hence \( R_{r_{ub}}(M, L, \hat{F}_r(M, K)) > R_{r_{ub}}(M, L, K) \). By noting that \( R_{MD}(M, L, \hat{F}_r(M, K)) \geq R_{m}(M, L, \hat{F}_r(M, K)) \) and \( R_{r_{ub}}(M, L, K) \geq R_r(M, L, K) \), we thus have \( R_{MD}(M, L, \hat{F}_r(M, K)) > R_r(M, L, K) \).
Proof of Inequality (6)

Let \( R_{tj}(M, L, \tilde{F}_{SD}(M, L, g), g) \) denote the worst-case load for serving the \( K' = \lceil \frac{1}{g} \cdot 3g \ln \left( \frac{1}{\gamma} \right) \rceil \) users in the \( j \)-th group. Note that \( R_{tj}(M, L, \tilde{F}_{SD}(M, L, g), g) \) is random. The worst-case load under Shanmugam et al.’s decentralized coded caching scheme is given by

\[
R_t(M, L, \tilde{F}_{SD}(M, L, g), g) = \mathbb{E} \left[ \sum_{j=1}^{L/K'} R_{tj}(M, L, \tilde{F}_{SD}(M, L, g), g) \right] = \frac{L}{K'} \mathbb{E} \left[ R_{tj}(M, L, \tilde{F}_{SD}(M, L, g), g) \right].
\]

(17)

Thus, to obtain a lower bound on \( R_t(M, L, \tilde{F}_{SD}(M, L, g), g) \) is equivalent to obtain a lower bound on \( \mathbb{E} \left[ R_{tj}(M, L, \tilde{F}_{SD}(M, L, g), g) \right] \). Let \( \mathcal{K}'_j \) denote the index set of the users in the \( j \)-th group. Let \( V_{k, \mathcal{S} \setminus \{k\}} \) denote the set of packets of file \( d_k \) stored in the cache of the users in set \( \mathcal{S} \setminus \{k\} \), after the “pull down phase” in Shanmugam et al.’s decentralized scheme. As the “pull down phase” brings the packets above type \( g \) to type \( g \), all the packets are present on type \( g \) or below [13]. Thus, we have

\[
\mathbb{E} \left[ R_{tj}(M, L, \tilde{F}_{SD}(M, L, g), g) \right] = \frac{\mathbb{E} \left[ \sum_{\mathcal{S} \subseteq \mathcal{K}'_j} \left| \mathcal{S} \cap \{s \leq g+1, k \in \mathcal{S}\} \right| \max_{k \in \mathcal{S}} |V_{k, \mathcal{S} \setminus \{k\}}| \right]}{\tilde{F}_{SD}(M, L, g)}
\]

\[
> \frac{\mathbb{E} \left[ \sum_{\mathcal{S} \subseteq \mathcal{K}'_j} \left| \mathcal{S} \cap \{s \leq g+1, k \in \mathcal{S}\} \right| \max_{k \in \mathcal{S}} |V_{k, \mathcal{S} \setminus \{k\}}| \right]}{\tilde{F}_{SD}(M, L, g)} = \frac{1}{\tilde{F}_{SD}(M, L, g)} \sum_{\mathcal{S} \subseteq \mathcal{K}'_j} \mathbb{E} \left[ \max_{k \in \mathcal{S}} |V_{k, \mathcal{S} \setminus \{k\}}| \right].
\]

(18)

Thus, to derive a lower bound on \( \mathbb{E} \left[ R_{tj}(M, L, \tilde{F}_{SD}(M, L, g), g) \right] \), we can derive a lower bound on \( \mathbb{E} \left[ \max_{k \in \mathcal{S}} |V_{k, \mathcal{S} \setminus \{k\}}| \right] \). Let \( Z_{n,i} \) denote the number of users who store packet \( i \) of file \( n \) before the “pull down phase”. Note that \( Z_{n,i} \) is random. Let \( \mathcal{B}_{n,g} \triangleq \{ i \in \{1, 2, \ldots, \tilde{F}_{SD}(M, L, g) \} : Z_{n,i} \geq g \} \) denote the set of packets of file \( n \), each of which is stored in no less than \( g \) users before the “pull down phase”. Note that \( \mathcal{B}_{n,g} \) is random, \( |\mathcal{B}_{n,g}| = \sum_{i=1}^{\tilde{F}_{SD}(M, L, g)} 1[Z_{n,i} \geq g] \in \{0, 1, \ldots, \tilde{F}_{SD}(M, L, g)\} \), and \( |\mathcal{B}_{n,g}|, n \in \mathcal{N} \) are independent. Denote \( \mathcal{B}_g \triangleq (|\mathcal{B}_{d_k,g}|)_{k' \in \mathcal{K}'} \in \{0, 1, \ldots, \tilde{F}_{SD}(M, L, g)\}^{K'} \). Let \( \mathbf{b} \) denote a \( K' \)-dimensional
vector with each element being $b \in \{0, 1, \cdots, \hat{F}_{SD}(M, L, g)\}$. Then, for any $b \in \{0, 1, \cdots, \hat{F}_{SD}(M, L, g)\}$, by [27, Corollary 1], we have

$$\mathbb{E}\left[\max_{k \in S}|V_{k, S\setminus\{k\}}|\right] \geq \mathbb{E}\left[\max_{k \in S}|V_{k, S\setminus\{k\}}|B_g = b\right] \prod_{k' \in K'_j} \Pr[|B_{d_{k', g}}| \geq b]. \quad (19)$$

By [26, Proposition 2], we have

$$\mathbb{E}\left[\max_{k \in S}|V_{k, S\setminus\{k\}}|B_g = b\right] \geq \mathbb{E}\left[|V_{k, S\setminus\{k\}}|B_g = b\right] = \frac{b}{K' g}. \quad (20)$$

When $\frac{1}{\gamma} \geq 8$, by using Chernoff bound, Markov’s inequality and choosing $b = \hat{F}_{SD}(M, L, g) - \left[\hat{F}_{SD}(M, L, g)\theta(M, g)\right]$, we have

$$\prod_{k' \in K'_j} \Pr[|B_{d_{k', g}}| \geq b] > 1 - \gamma. \quad (21)$$

By (19), (21) and (20), when $\frac{1}{\gamma} \geq 8$, we have

$$\mathbb{E}\left[\max_{k \in S}|V_{k, S\setminus\{k\}}|\right] > \frac{b}{K' g} (1 - \gamma). \quad (22)$$

By (18) and (22), when $\frac{1}{\gamma} \geq 8$, we have

$$\mathbb{E}\left[R_{tj}(M, L, \hat{F}_{SD}(M, L, g), g)\right] = \frac{K' - g}{g + 1} \left(1 - \frac{\hat{F}_{SD}(M, L, g)\theta(M, g)}{\hat{F}_{SD}(M, L, g)}\right) (1 - \gamma). \quad (23)$$

By (17) and (23), when $\frac{1}{\gamma} \geq 8$, we have $R_t(M, L, \hat{F}_{SD}(M, L, g), g) = \frac{K'}{K''} \mathbb{E}\left[R_{tj}(M, L, \hat{F}_{SD}(M, L, g), g)\right] > \frac{L}{g + 1} c(M, \hat{F}_{SD}(M, L, g), g)$. Therefore, we complete the proof of (6).

**Proof of Inequality (7)**

To prove (7), we first derive another lower bound on $\mathbb{E}\left[R_{tj}(M, L, \hat{F}_{SD}(M, L, g), g)\right]$. By (18) and (19), we know that to derive a lower bound on $\mathbb{E}\left[R_{tj}(M, L, \hat{F}_{SD}(M, L, g), g)\right]$, we can derive a lower bound on $\prod_{k' \in K'_j} \Pr[|B_{d_{k', g}}| \geq b]$ and a lower bound on $\mathbb{E}\left[\max_{k \in S}|V_{k, S\setminus\{k\}}|B_g = b\right]$, separately. Here, we use the lower bound on $\prod_{k' \in K'_j} \Pr[|B_{d_{k', g}}| \geq b]$ given by (21). It remains to derive a new lower bound
on \( E \left[ \max_{k \in S} |V_{k,S\setminus\{k\}}| \bigg| B_g = b \right] \). We consider \( S \in \left\{ \hat{S} \subseteq \mathcal{K}_j \big| |\hat{S}| = g + 1, k \in \hat{S} \right\} \). By conditional Markov's inequality and Bonferroni inequality, we have

\[
E \left[ \max_{k \in S} |V_{k,S\setminus\{k\}}| \bigg| B_g = b \right] \geq \frac{(g+1)b}{K_g} \left( 1 - \frac{(g+1)b}{K_g} \right) \tag{24}
\]

By (19), (21) and (24), we have

\[
E \left[ \max_{k \in S} |V_{k,S\setminus\{k\}}| \bigg| B_g = b \right] > \frac{(g+1)b}{g+1} \frac{(g+1)b}{K_g} \left( 1 - \frac{(g+1)b}{K_g} \right) (1 - \gamma) \tag{25}
\]

By (18) and (25), we have

\[
E \left[ R_{ij}(M, L, \hat{F}_{SD}(M, L, g), g) \right] > \left( \frac{K'}{g+1} \right) \frac{(g+1)b}{\hat{F}_{SD}(M, L, g)\left( \frac{K'}{g} \right)} \left( 1 - \frac{(g+1)b}{\hat{F}_{SD}(M, L, g)\left( \frac{K'}{g} \right)} \right) (1 - \gamma) \tag{26}
\]

By (26) and (17), we can obtain inequality (7).

**APPENDIX D: PROOF OF THEOREM 3**

By (6), (7) as well as \( \binom{n}{k} \geq \binom{n}{k} \) for all \( n, k \in \mathbb{N} \) and \( n \geq k \), we have

\[
\hat{F}_{SD}(M, L, g) > \left( 1 - \frac{R_{SD}(M, \hat{F}_{SD}(M, L, g), g, L)}{L (1 - \gamma) \left( 1 - \frac{g}{K} \right) \left( 1 - \frac{\hat{F}_{SD}(M, L, g)\theta(M, g)}{\hat{F}_{SD}(M, L, g)} \right)} \right) \frac{\ln \left( \frac{1}{\gamma} \right)}{g+1} \frac{\left( \frac{1}{\gamma} \right)}{(1 - \frac{\hat{F}_{SD}(M, L, g)\theta(M, g)}{\hat{F}_{SD}(M, L, g)})} \tag{27}
\]

By (27), when \( R_{SD}(M, L, \hat{F}_{SD}(M, L, g), g) = R_r(M, L, K) \), we have

\[
\frac{\hat{F}_{SD}(M, L, g)}{\hat{F}_r(M, K)} > \left( 1 - \frac{R_r(M, L, K)}{L (1 - \gamma) \left( 1 - \frac{g}{K} \right) \left( 1 - \frac{\hat{F}_{SD}(M, L, g)\theta(M, g)}{\hat{F}_{SD}(M, L, g)} \right)} \right) \frac{\ln \left( \frac{1}{\gamma} \right)}{g+1} \frac{\left( \frac{1}{\gamma} \right)}{(1 - \frac{\hat{F}_{SD}(M, L, g)\theta(M, g)}{\hat{F}_{SD}(M, L, g)})} \tag{28}
\]

By (28), we have \( \lim_{(L, \frac{1}{\gamma}) \to (\infty, \infty)} \frac{\hat{F}_{SD}(M, L, g)}{\hat{F}_r(M, K)} \to \infty \). Thus, we know that, at the same given worst-case load, there exists \( \tilde{L}_r > 0 \) and \( q_r > 0 \), such that when \( L > \tilde{L}_r \) and \( \frac{1}{\gamma} > q_r \), we have \( \hat{F}_{SD}(M, L, g) > \hat{F}_r(M, K) \).
APPENDIX E: PROOF OF LEMMA 4

First, we show that Pr[\(\hat{K}_{1} = L\)] \(\to\) 1, as \(K \to \infty\). Note that \(\hat{K}_{1} = L\) if and only if \(X_{k} \in \{0, 1\}\) for all \(k \in \mathcal{K}\). Thus, \(\hat{K}_{1} = L\) and \(x \in X_{Ld} \triangleq \left\{ (x_{1}, x_{2}, \ldots, x_{K}) : \sum_{k=1}^{K} x_{k} = L, x_{k} \in \{0, 1\} \right\} \subset X_{K,L}\) imply each other. When \(K \geq L\), we have

\[
\lim_{K \to \infty} \Pr[\hat{K}_{1} = L] = \lim_{K \to \infty} \sum_{x \in X_{Ld}} P_{X}(x) \overset{(a)}{=} \sum_{x \in X_{Ld}} \frac{L!}{K^{L}} \overset{(b)}{=} \lim_{K \to \infty} \left( \frac{K}{L} \right) \frac{L!}{K^{L}} = 1, \tag{29}
\]

where (a) is due to \(P_{X}(x) = \frac{L!}{x_{1}!x_{2}! \cdots x_{K}!} \frac{1}{K^{L}}\) for all \(x \in X_{Ld}\), and (b) is due to \(|X_{Ld}| = \binom{K}{L}\).

Then, we show that \(R_{\infty}(M, L) = \left(\frac{1}{\gamma} - 1\right) \left(1 - (1 - \gamma)^{L}\right)\). Denote \(\overline{X}_{Ld} \triangleq X_{K,L} \setminus X_{Ld}\), \(R_{d}(M, L, K) \triangleq \sum_{x \in \overline{X}_{Ld}} P_{X}(x)R(M, L, K, x)\), and \(R_{\overline{d}}(M, L, K) \triangleq \sum_{x \in X_{Ld}} P_{X}(x)R(M, L, K, x)\). We have \(R_{r}(M, L, K) = R_{d}(M, L, K) + R_{\overline{d}}(M, L, K)\). To calculate \(R_{r}(M, L, K)\), we calculate \(\lim_{K \to \infty} R_{d}(M, L, K)\) and \(\lim_{K \to \infty} R_{\overline{d}}(M, L, K)\), respectively.

1) When \(K \geq \frac{L+1}{1-\gamma}\), by Pascals identity, we have

\[
R(M, L, K, x) = \frac{\binom{K}{K+1} - \binom{K-1}{K+1}}{\frac{L}{K}} = \left(\frac{1}{\gamma} - 1\right) \left(1 - (1 - \gamma)^{L}\right), \quad x \in X_{Ld}. \tag{30}
\]

Taking limits of both sides of (30), we have

\[
\lim_{K \to \infty} R(M, L, K, x) = \lim_{K \to \infty} \left(1 - \frac{\gamma}{1+K} \right) \left(1 - (1 - \gamma)^{L}\right) = \left(\frac{1}{\gamma} - 1\right) \left(1 - (1 - \gamma)^{L}\right), \quad x \in X_{Ld}. \tag{31}
\]

Thus, we have \(\lim_{K \to \infty} R_{d}(M, L, K) \overset{(c)}{=} \lim_{K \to \infty} R(M, L, K, x) \sum_{x \in X_{Ld}} P_{X}(x) \overset{(d)}{=} \left(\frac{1}{\gamma} - 1\right) \left(1 - (1 - \gamma)^{L}\right)\), where (c) is due to the fact that when \(K \geq \frac{L+1}{1-\gamma}\), the values of \(R(M, L, K, x), x \in X_{Ld}\) are the same, and (d) is due to (29) and (31).

2) As \(\lim_{K \to \infty} \sum_{x \in \overline{X}_{Ld}} P_{X}(x) = 0\), we have \(\lim_{K \to \infty} R_{\overline{d}}(M, L, K) = \lim_{K \to \infty} R(M, L, K, x) \sum_{x \in \overline{X}_{Ld}} P_{X}(x) = 0\).

Thus, by 1) and 2), we have \(R_{r}(M, L) = \lim_{K \to \infty} R_{d}(M, L, K) + \lim_{K \to \infty} R_{\overline{d}}(M, L, K) = \left(\frac{1}{\gamma} - 1\right) \left(1 - (1 - \gamma)^{L}\right)\).

Next, we derive the asymptotic approximation of an upper bound on \(R_{r}(M, L, K)\), i.e., \(\overline{R}_{r}^{ub}(M, L, K) \triangleq \sum_{x \in X_{Ld}} P_{X}(x)R(M, L, K, x) + \sum_{x \in \overline{X}_{Ld}} P_{X}(x)L(1-\gamma)\). By asymptotic analysis, we can show that \(\overline{R}_{r}^{ub}(M, L, K) = R_{r}(M, L) + \frac{\overline{A}(M,L)}{K} + o\left(\frac{1}{K}\right)\), as \(K \to \infty\) [27]. Thus, we can obtain (10).

Finally, we show \(A(M, L) \geq 0\). Denote \(g(z, L) \triangleq (1-z)^{(L-1)}(1+\frac{(L+2)(L-1)}{2}z) - 1 + \frac{L(L-1)}{2}z(Lz-1)\) and \(h(z, L) \triangleq \frac{L^2+3L}{2}(1-z)L - \frac{L^2+L}{2}(1-z)^{L-1} - L + z\frac{L^2+L}{2}\). Note that \(A(M, L) = \frac{1}{\gamma}(\frac{1}{\gamma} - 1)g(\gamma, L), g(\gamma, 2) = 0,\)
\[ g(\gamma, L + 1) - g(\gamma, L) = \gamma h(\gamma, L) \] and \( \frac{\partial h(z, L)}{\partial z} > 0 \) for all \( z \in (0, 1) \) and \( L \in \{2, 3, \ldots\} \). Thus, when \( L \in \{2, 3, \ldots\} \), we have \( g(\gamma, L + 1) - g(\gamma, L) > \gamma h(0, L) = 0 \), implying that \( A(M, L) \geq \frac{1}{\gamma} (\frac{1}{\gamma} - 1) g(\gamma, 2) = 0 \).

Therefore, we complete the proof of Lemma 4.

**APPENDIX E: PROOF OF THEOREM 4**

**Proof of Statement (i)**

First, we prove \( 1 \leq G_r(M, L, K) < 1 + K\gamma \). As \( R_r(M, L, K) \leq L(1 - \gamma) \), we have \( G_r(M, L, K) = \frac{R_u(M, L)}{R_r(M, L, K)} \geq 1 \). In addition, by [27, Theorem 3], we have \( R_r(M, L, K) > \frac{L(1 - \gamma)}{1 + K\gamma} \). Thus, we have
\[
G_r(M, L, K) = \frac{R_u(M, L)}{R_r(M, L, K)} < \frac{L(1 - \gamma)}{L(1 - \gamma) + K\gamma} = 1 + K\gamma.
\]
(32)

Next, we prove \( \lim_{L \to \infty} G_r(M, L, K) = 1 + K\gamma \). By (15), we have
\[
G_r(M, L, K) \geq \frac{L(1 + K\gamma)}{L + K\sqrt{\frac{L}{2}}} \to 1 + K\gamma, \quad \text{as } L \to \infty.
\]
(33)

By (32) and (33), we have \( \lim_{L \to \infty} G_r(M, L, K) = 1 + K\gamma \).

**Proof of Statement (ii)**

First, we prove \( \hat{F}_r(M, K) \geq \left(\frac{1}{e}\right)^{G_r(M, L, K) - 1} \). By (32), we have \( K > \frac{1}{\gamma} (G_r(M, L, K) - 1) \). By noting that \( \left(\frac{n}{k}\right) \geq \left(\frac{n}{k}\right)^k \) for all \( n, k \in \mathbb{N} \) and \( n \geq k \), we have \( \hat{F}_r(M, K) = \left(\frac{K}{K\gamma}\right) \geq \left(\frac{1}{e}\right)^{G_r(M, L, K) - 1} \).

Next, we prove \( \hat{F}_r(M, K) \leq \left(\frac{1}{e}\right)^{\frac{(G_r(M, L, K) - 1)\sqrt{2L}}{\sqrt{2L} - G_r(M, L, K) \frac{\gamma^2}{4}}} \). When \( L \in \left\{\left[\frac{1}{2}\left(\frac{1}{\gamma}\right)^2\right], \left[\frac{1}{2}\left(\frac{1}{\gamma}\right)^2\right] + 1, \ldots\right\} \) and \( G_r(M, L, K) \in \left(1, \min\left\{\frac{\sqrt{2LM}}{N}, 1 + K\gamma\right\}\right) \), by (15), we have \( K \leq \frac{G_r(M, L, K) - 1}{\sqrt{2L - G_r(M, L, K) \frac{\gamma^2}{4}}} \). By noting that \( \left(\frac{n}{k}\right) \leq \left(\frac{n}{k}\right)^k \) for all \( n, k \in \mathbb{N} \) and \( n \geq k \), we have \( \hat{F}_r(M, K) \leq \left(\frac{K}{K\gamma}\right) \leq \left(\frac{1}{e}\right)^{G_r(M, L, K) - 1} \), for all \( G_r(M, L, K) \in \left(1, \min\left\{\frac{\sqrt{2LM}}{N}, 1 + K\gamma\right\}\right) \).

**APPENDIX G: PROOF OF LEMMA 5**

By (9), we can easily obtain (12). It remains to prove (13). By (10), we have
\[
G_r(M, L, K) = \frac{R_u(M, L)}{R_r(M, L, K)} = \frac{R_u(M, L)}{R_\infty(M, L) + \frac{A(M, L)}{K} + o\left(\frac{1}{K}\right)} = \frac{R_u(M, L)}{R_\infty(M, L) + \frac{A(M, L)}{K} + o\left(\frac{1}{K}\right)}
\]
(34)
where (a) is due to \( \frac{1}{1+x} = 1 - x + o(x) \) as \( x \to 0 \). By Stirling’s approximation, when \( K \) is large, we have
\[
\frac{1}{K} = \frac{H(\frac{1}{2})}{\ln \hat{F}_r(M, K)} + o \left( \frac{1}{\ln \hat{F}_r(M, K)} \right).
\]
Substituting (35) into (34), we can obtain (13).

REFERENCES

[1] Cisco, “Cisco visual networking index: Global mobile data traffic forecast update, 2016 - 2021,” March 2017.
[2] K. Shanmugam, N. Golrezaei, A. G. Dimakis, A. F. Molisch, and G. Caire, “Femtocaching: Wireless content delivery through distributed caching helpers,” *IEEE Trans. Inf. Theory*, vol. 59, no. 12, pp. 8402–8413, Dec 2013.
[3] K. Poularakis, G. Iosifidis, and L. Tassiulas, “Approximation algorithms for mobile data caching in small cell networks,” *IEEE Trans. Commun.*, vol. 62, no. 10, pp. 3665–3677, Oct 2014.
[4] A. Liu and V. K. N. Lau, “Exploiting base station caching in mimo cellular networks: Opportunistic cooperation for video streaming,” *IEEE Trans. Signal Process.*, vol. 63, no. 1, pp. 57–69, Jan 2015.
[5] B. Zhou, Y. Cui, and M. Tao, “Stochastic content-centric multicast scheduling for cache-enabled heterogeneous cellular networks,” *IEEE Trans. Wireless Commun.*, vol. 15, no. 9, pp. 6284–6297, Sept 2016.
[6] D. Lecompte and F. Gabin, “Evolved multimedia broadcast/multicast service (embms) in lte-advanced: overview and rel-11 enhancements,” *IEEE Commun. Mag.*, vol. 50, no. 11, pp. 68–74, November 2012.
[7] K. Poularakis, G. Iosifidis, V. Sourlas, and L. Tassiulas, “Multicast-aware caching for small cell networks,” in *IEEE WCNC*, April 2014, pp. 2300–2305.
[8] N. Abedini and S. Shakkottai, “Content caching and scheduling in wireless networks with elastic and inelastic traffic,” *IEEE/ACM Trans. Netw.*, vol. 22, no. 3, pp. 864–874, June 2014.
[9] Y. Cui, D. Jiang, and Y. Wu, “Analysis and optimization of caching and multicasting in large-scale cache-enabled wireless networks,” *IEEE Trans. Wireless Commun.*, vol. 15, no. 7, pp. 5101–5112, July 2016.
[10] Y. Cui and D. Jiang, “Analysis and optimization of caching and multicasting in large-scale cache-enabled heterogeneous wireless networks,” *IEEE Trans. Wireless Commun.*, vol. 16, no. 1, pp. 250–264, Jan 2017.
[11] Y. Cui, Z. Wang, Y. Yang, F. Yang, L. Ding, and L. Qian, “Joint and competitive caching designs in large-scale multi-tier wireless multicasting networks,” *IEEE Transactions on Communications*, pp. 1–1, 2018.
[12] M. A. Maddah-Ali and U. Niesen, “Fundamental limits of caching,” *IEEE Trans. Inf. Theory*, vol. 60, no. 5, pp. 2856–2867, May 2014.
[13] K. Shanmugam, M. Ji, A. M. Tulino, J. Llorca, and A. G. Dimakis., “Finite-length analysis of caching-aided coded multicasting,” *IEEE Trans. Inf. Theory*, vol. 62, no. 10, pp. 5524–5537, Oct. 2016.
[14] M. A. Maddah-Ali and U. Niesen, “Decentralized coded caching attains order-optimal memory-rate tradeoff,” *IEEE/ACM Trans. Netw.*, vol. 23, no. 4, pp. 1029–1040, Aug. 2015. [Online]. Available: https://doi.org/10.1109/TNET.2014.2317316
[15] A. Ramakrishnan, C. Westphal, and A. Markopoulou, “An efficient delivery scheme for coded caching,” in *2015 27th International Teletraffic Congress*, Sept 2015, pp. 46–54.
[16] K. Wan, D. Tuninetti, and P. Plantanida, “Novel delivery schemes for decentralized coded caching in the finite file size regime,” in *IEEE ICC Workshops*, May 2017, pp. 1183–1188.
[17] M. Ji, K. Shanmugam, G. Vettigli, J. Llorca, A. M. Tulino, and G. Caire, “An efficient multiple-groupcast coded multicasting scheme for finite fractional caching,” in IEEE ICC, June 2015, pp. 3801–3806.

[18] Q. Yan, X. Tang, and Q. Chen, “On the placement and delivery schemes for decentralized coded caching system,” CoRR, vol. abs/1710.04884, 2017.

[19] U. Niesen and M. A. Maddah-Ali, “Coded caching with nonuniform demands,” IEEE Trans. Inf. Theory, vol. 63, no. 2, pp. 1146–1158, Feb 2017.

[20] J. Zhang, X. Lin, and X. Wang, “Coded caching under arbitrary popularity distributions,” IEEE Trans. Inf. Theory, vol. 64, no. 1, pp. 349–366, Jan 2018.

[21] S. Wang, X. Tian, and H. Liu, “Exploiting the unexploited of coded caching for wireless content distribution,” in IEEE ICNC, Feb 2015, pp. 700–706.

[22] R. Pedarsani, M. A. Maddah-Ali, and U. Niesen, “Online coded caching,” IEEE/ACM Trans. Netw., vol. 24, no. 2, pp. 836–845, Apr 2016.

[23] M. Ji, A. M. Tulino, J. Llorca, and G. Caire, “Order-optimal rate of caching and coded multicasting with random demands,” IEEE Trans. Inf. Theory, vol. 63, no. 6, pp. 3923–3949, June 2017.

[24] S. Jin, Y. Cui, H. Liu, and G. Caire, “Structural properties of uncoded placement optimization for coded delivery,” CoRR, vol. abs/1707.07146, 2017.

[25] Q. Yu, M. A. Maddah-Ali, and A. S. Avestimehr, “The exact rate-memory tradeoff for caching with uncoded prefetching,” IEEE Trans. Inf. Theory, vol. 64, no. 2, pp. 1281–1296, Feb 2018.

[26] O. Gascuel and G. Caraux, “Bounds on expectations of order statistics via extremal dependences,” Statistics & probability letters, vol. 15, no. 2, pp. 143–148, 1992.

[27] S. Jin, Y. Cui, H. Liu, and G. Caire, “New order-optimal decentralized coded caching schemes with good performance in the finite file size regime,” CoRR, vol. abs/1604.07648, 2016. [Online]. Available: http://arxiv.org/abs/1604.07648