Layered Caching for Heterogeneous Storage

Avik Sengupta
Wireless@VT & Hume Center,
Department of ECE, Virginia Tech,
Blacksburg, VA 24060, USA
Email: aviksg@vt.edu

Ravi Tandon
Department of ECE
University of Arizona,
Tucson, AZ 85721, USA
Email: tandonr@email.arizona.edu

T. Charles Clancy
Hume Center
Department of ECE, Virginia Tech,
Arlington, VA 22203 USA
Email: tcc@vt.edu

***Abstract*—In modern data-centric wireless networks, caching alleviates severe capacity crunch at times of high network load. Careful design of cache storage to leverage coded multicast file delivery over a shared link can achieve order-wise improvement in delivery rates. Recent results have shown that careful design of cache storage to leverage coded multicast file delivery over a shared link can achieve order-wise improvements in the storage-rate trade-off. In this work, we present a novel caching and delivery scheme for the case when users have *heterogeneous cache sizes*. The proposed scheme uses two new ingredients namely set partitioning and cache layering. The main challenge in designing caching schemes in presence of storage heterogeneity is that varying levels of storage across users can present a variety of storage/multicasting opportunities. Our framework of cache layering and set partitioning is a principled approach to utilize such opportunities, where each layer delivers a fraction of requested data to a specific set of users and layers operate independently of each other. We also derive an information-theoretic lower bound for the heterogeneous caching problem.**

I. Introduction

Caching has emerged as an important tool for efficient load balancing in future content-centric 5G wireless networks. Parts of popular content can be sub-packetized and pre-stored in end users' device caches such that the local content can be leveraged to reduce the over-the-air transmission rates at times of peak network loads. Caching operates in two phases - (1) the *storage phase* where parts of popular content is pre-fetched and stored in users' device caches and (2) the *delivery phase*, where requested content is delivered by exploiting the local cache storage. Consider a caching system with K users and a server which has a library of N files (denoted by $(F_1, F_2, \ldots, F_N)$, each of size B bits), we note first, that for *homogeneous caching*, each user $k \in 1, \ldots, K$, has a cache storage of size MB bits. In the storage phase, the caches of the users are populated with some function of files Z_k. Once the user demands are revealed, the server delivers content at a rate of RB bits via a shared link to the users. The received transmission in conjunction with the user cache content Z_k is capable of decoding the requested file at each user. The fundamental tradeoff for the homogeneous caching system is that of per-user cache storage vs. transmission rate (referred to as (M, R) tradeoff).

For the homogeneous caching problem, recent results by Maddah-Ali and Niesen [1]–[3] show that by jointly designing the storage and delivery phase, and using multicast transmissions to simultaneously deliver content to users, order-wise improvement in the (M, R) tradeoff can be achieved compared to traditional point-to-point unicast delivery. Improved lower bounds for homogeneous caching were presented in [4]–[6] in addition to several extensions e.g. caching for non-uniform file popularities [3], [7], hierarchical caching [8], secure delivery [9] and extensions to device-to-device systems [10], [11]. A common underlying assumption in these works is that each user is equipped with the same cache storage. In practice, however, different types of devices and users in the system might possess different storage capabilities which motivates the study of a network with *heterogeneous cache storage*.

The main contributions of this work are as follows. We introduce the heterogeneous caching problem where each user $k \in \{1, 2, \ldots, K\}$, has a potentially different cache storage of size $M_k B$ bits. For the heterogeneous caching problem, we introduce a scheme with two key ingredients: *set partitioning* and *cache layering*. We first partition the set of K users into disjoint sub-sets, where each sub-set of users is dealt with separately. For each sub-set of users, we propose a layered caching scheme which works as follows: each layer is dedicated to the storage/delivery of a specific fraction of the files, and this fraction is selected based on the level of storage heterogeneity within the users in the sub-set. We show that the proposed scheme provides significant improvements over the naive extensions of the homogeneous scheme presented in [1] to the heterogeneous case. We derive an information theoretic lower bound on the optimal rate of the heterogeneous caching problem and show that the proposed scheme is order optimal for systems with 2 and 3 levels of heterogeneity. We highlight interesting aspects of the impact of heterogeneity on the achievable rate of the proposed scheme such as the reduction of multicasting gain and usefulness of set partitioning with increase in heterogeneity.

II. System Model

We consider a system with K users with each user $k \in \{1, \ldots, K\}$ having a cache storage of $M_k B$ bits, for some $B \in \mathbb{N}^+$. Furthermore, we define an ordered set of K heterogeneous caches $\mathcal{M} := \{M_1, M_2, \ldots, M_K\}$ where $M_1 \leq M_2 \leq \ldots \leq M_K$. The network has a library of N files, $F_1, \ldots, F_N$, each of size B bits. Formally, the files F_n are independent and identically distributed (i.i.d.) as:

$$F_n \sim \text{Unif}\{1, 2, 3, \ldots, 2^B\}, \quad \forall n = 1, \ldots, N. \tag{1}$$

Fig. 1 illustrates the system model. We next define the key components of the heterogeneous caching problem. The storage phase consists of K caching functions, which map the files $(F_1, \ldots, F_N)$ into the cache content

$$Z_k \triangleq \phi_k\big(F_1, \ldots, F_N\big), \tag{2}$$

for each user k. The maximum allowable size of each user's cache content Z_k is $M_k B$ such that $H(Z_k) \leq M_k B$. The

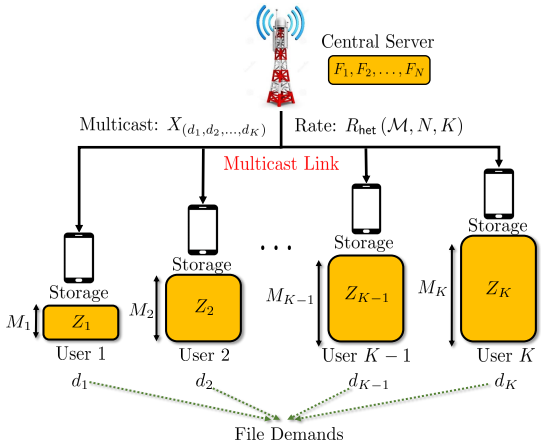


Fig. 1. System Model for Caching with Heterogeneous Storage.

content delivery phase consists of N^K encoding functions which map files $(F_1, \ldots, F_N)$ to the transmission

$$X_{(d_1,\ldots,d_K)} \triangleq \psi_{(d_1,\ldots,d_K)}(F_1, \ldots, F_N), \tag{3}$$

over the shared link to service the user demands $(d_1, \ldots, d_K) \in \{1, 2, 3, \ldots, N^K\}$. Each such transmission has a rate not exceeding $R_{\text{het}}B$ bits. Once the transmission is received, KN^K decoding functions map the received signal over the shared link $X_{(d_1,\ldots,d_K)}$ and the cache content Z_k to the estimate

$$\hat{F}_{d_k} \triangleq \mu_{(d_1,\ldots,d_K),k}\Big(X_{(d_1,\ldots,d_K)}, Z_k\Big), \tag{4}$$

of the requested file F_{d_k} for user k. For a $(\mathcal{M}, R_{\text{het}})$ caching scheme, the worst-case probability of error is defined as:

$$P_e \triangleq \max_{(d_1,\ldots,d_K)\in\{1,\ldots,N^K\}} \max_{k\in\{1,\ldots,K\}} \mathbb{P}(\hat{F}_{d_k} \neq F_{d_k}). \tag{5}$$

Definition 1 (*Storage vs. Rate Tradeoff*)**.** The storage-rate pair $(\mathcal{M}, R_{\text{het}})$ is *achievable* if, for any $\epsilon > 0$, there exists an $(\mathcal{M}, R_{\text{het}})$ caching scheme for which $P_e \leq \epsilon$. The optimal storage vs. rate tradeoff is defined as:

$$R^*_{\text{het}}(\mathcal{M}, N, K) \triangleq \inf\{R_{\text{het}} : (\mathcal{M}, R_{\text{het}}) \text{ is achievable}\}. \tag{6}$$

III. Existing Results and Preliminaries

A. Existing Results for Homogeneous Caching

For N files and K users with homogeneous cache storage where $M_1 = M_2 = \ldots = M_K = M$ such that $M \in (0, N]$, a novel caching and coded delivery scheme was presented in [1] which yields an achievable rate of $R_{\text{hom}}(M, N, K) = \min\left\{R^u_{\text{hom}}(M, N, K), R^m_{\text{hom}}(M, N, K)\right\}$, which is a minimum of the conventional unicast rate

$$R^u_{\text{hom}}(M, N, K) = \min\{N, K\}\left(1 - \frac{M}{N}\right), \tag{7}$$

where each user's demand is served with individual point-to-point transmissions and the multicast rate

$$R^m_{\text{hom}}(M, N, K) = \frac{1}{t+2}\left[2K - t - \frac{K+1}{t+1}\cdot\frac{KM}{N}\right] \tag{8}$$

with $t = \lfloor \frac{KM}{N} \rfloor$, where multiple users' demands are jointly serviced with multicast transmissions. The multicast delivery achieves a global caching gain which leads to order-wise improvements in the storage-rate trade-off compared to traditional unicast delivery.

B. Naive Extensions to Heterogeneous Setting

We next introduce the preliminaries for heterogeneous caching by extending known homogeneous schemes to the heterogeneous setting.

• *Heterogeneous Unicast*: Let users cache bits of files in a sequential manner i.e., user k caches the first $\frac{M_k B}{N}$ bits from each of the N files. Under this sequential caching, it is enough to deliver the largest complementary fragments of common requested content. Given the set of ordered caches $\mathcal{M}$, the heterogeneous unicast rate is given by:

$$R^u_{\text{het}}(\mathcal{M}, N, K) = \min\{N, K\} - \frac{\sum_{i=1}^{\min\{N,K\}} M_i}{N}. \tag{9}$$

• *Heterogeneous Multicast*: Given a set of caches $\mathcal{M}$, the storage and delivery is designed based on the lowest cache storage in the set and the achievable rate in this case is given by

$$R^m_{\text{het}}(\mathcal{M}, N, K) = R^m_{\text{hom}}\left(\min_{M_i}\{\mathcal{M}\}, N, K\right), \tag{10}$$

where $\min\{\mathcal{M}\}$ is the smallest storage in $\mathcal{M}$ and R^m_{hom} is given in (8).

Considering the heterogeneous unicast scheme, although every user's storage is completely utilized, the transmissions are point-to-point and the global caching gain due to multicast is lost. Conversely, the naive multicasting scheme is limited by the lowest storage in the system. Further, as heterogeneity increases, the strategy leads to *cache wastage* i.e., for any user k, $(M_k - \min\{\mathcal{M}\})$ amount of storage is not utilized. The ideal heterogeneous caching and delivery scheme should combine the complete utilization of each users' storage while also leveraging multicasting opportunities. This forms the basis of the proposed *Layered Heterogeneous Caching* (LHC) scheme discussed in the sequel.

IV. Main Results and Discussion

In this section we present new upper and lower bounds on the optimal rate for heterogeneous caching. The following theorem gives our main result, which is an upper bound on the optimal rate, $R^*_{\text{het}}(\mathcal{M}, N, K)$, of the heterogeneous caching problem based on the proposed Layered Heterogeneous Caching (LHC) scheme.

Theorem 1. *For any N files and K users with heterogeneous cache storage $\mathcal{M} := \{M_1, M_2, \ldots, M_K\} \in (0, N]$,*

$$R^*_{\text{het}}(\mathcal{M}, N, K) \leq \min_{\mathcal{G}\in\mathcal{P}_K} \sum_{g\in\mathcal{G}} R_{\text{het}}(\mathcal{M}_g, N, K_g), \tag{11}$$

where $\mathcal{P}_K$ is the set of all possible partitions of the set of caches $\mathcal{M}$ and $\mathcal{G} \in \mathcal{P}_K$, with cardinality $G = |\mathcal{G}|$, is any valid partitioning of $\mathcal{M}$ into non-overlapping ordered subsets $\mathcal{M}_g \subseteq \mathcal{M}$ with K_g users for $g \in \{1, 2, \ldots, G\}$. The achievable rate for cache set $\mathcal{M}_g$ is given by

$$R_{\text{het}}(\mathcal{M}_g, N, K_g) = \begin{cases} R^u_{\text{het}}(\mathcal{M}_g, N, K_g), & \text{if } K_g = 1 \\ R^{\text{LHC}}_{\text{het}}(\mathcal{M}_g, N, K_g, \vec{\alpha}^*_g), & \text{if } K_g > 1 \end{cases} \tag{12}$$

where $R^{\text{LHC}}_{\text{het}}$ is the rate achieved by the LHC scheme outlined in Algorithm 1 and $\vec{\alpha}^*_g = \left\{\alpha^*_1, \ldots, \alpha^*_{K_g}\right\}$ is the optimal file splitting strategy for any ordered cache set $\mathcal{M}_g$.

The achievable rate in Theorem 1 results from two main concepts namely, (i) *set partitioning* of the ordered cache set $\mathcal{M}$ and (ii) *cache layering* for every ordered subset of heterogeneous caches in a given partition. We next elaborate on these two strategies and outline the proposed LHC scheme detailed in Algorithm 1.

A. Set Partitioning

Set partitioning is used to determine the best grouping of caches in order to maximize the achievable rate of the heterogeneous caching scheme. Let the cache set $\mathcal{M}$ be *partitioned* [12] into a set of disjoint subsets. Let one such partition be $\mathcal{G} = \{\mathcal{M}_1, \mathcal{M}_2, \ldots, \mathcal{M}_G\}$ for some integer $G = |\mathcal{G}| \leq K$, such that $\mathcal{M}_g \subseteq \mathcal{M}$ is an ordered subset of K_g users' caches $\forall g \in \{1, 2, \ldots, G\}$. For any valid partitioning $\mathcal{G} \in \mathcal{P}_K$ we also have $\{\mathcal{M}_i \cap \mathcal{M}_j\} = \emptyset$ for $i \neq j,\ \forall i, j \in \{1, 2, \ldots, G\}$ and $\bigcup_{g=1}^{G} \mathcal{M}_g = \mathcal{G}$. Based on such a partitioning, for any subset of caches $\mathcal{M}_g$, if $K_g = 1$, i.e., only one user is present in the group $\mathcal{M}_g$, sequential caching and unicast transmission is used to deliver content to this user. For all other groups $\mathcal{M}_g$ such that $K_g \geq 2$, multicasting opportunities can be exploited. The following example illustrates the concept of partitioning.

Example 1. For a system with N files and $K = 5$ users with cache set $\mathcal{M} := \{M_1, M_2, \ldots, M_5\}$, a valid partition can be $\mathcal{G} = \{\{M_1\}, \{M_2, M_3\}, \{M_4, M_5\}\}$. Here $G = 3$ and the ordered subsets $\mathcal{M}_1 = \{M_1\}$, $\mathcal{M}_2 = \{M_2, M_3\}$, $\mathcal{M}_3 = \{M_4, M_5\}$. Based on this partition, unicast delivery is used for the group $\mathcal{M}_1$, while for $\mathcal{M}_2$ and $\mathcal{M}_3$, the storage and delivery is based on the proposed LHC scheme. □

The proposed LHC scheme is a principled approach to exploit all possible multicast and unicast opportunities for a given subset of caches $\mathcal{M}_g$ for any partitioning of $\mathcal{M}$. Note that, when $G = 1$, all users are grouped together and served via LHC, while for $G = K$, every user is served individually via unicast. For K users, consider the set of all possible partitions $\mathcal{P}_K$, of the cache set $\mathcal{M}$. The total number of such partitions i.e., the cardinality of $\mathcal{P}_K$ is given by the K-th Bell Number [12]. Minimizing over the achievable unicast and LHC rates for all possible partitions yields an upper bound on the optimal rate in (11). The optimal partition $\mathcal{G}^{\mathsf{opt}}$ is then given by

$$\mathcal{G}^{\mathsf{opt}} = \underset{\mathcal{G} \in \mathcal{P}_K}{\arg\min} \quad \sum_{g \in \mathcal{G}} R_{\mathsf{het}}\left(\mathcal{M}_g, N, K_g\right). \tag{13}$$

B. Layered Heterogeneous Caching (LHC) Scheme

In this section, we propose the Layered Heterogeneous Caching (LHC) scheme based on a novel cache layering and file splitting strategy as follows:

• *Cache Layering*: For any partitioning of the cache set $\mathcal{M}$, consider the subset of ordered caches $\mathcal{M}_g$ with K_g users. The caches in $\mathcal{M}_g$ are then divided into K_g *layers*, $\mathcal{L}_1, \ldots, \mathcal{L}_{K_g}$. Layer $\mathcal{L}_1$ consists of all K_g users, each with a storage of $\mathsf{m}_1 = M_1$. Layer $\mathcal{L}_2$ consists of users 2 to K_g (i.e., $K_g - 1$ users) each with a cache storage of $\mathsf{m}_2 = (M_2 - M_1)$. In general, layer $\mathcal{L}_\ell,\ \forall \ell \in \{1, 2, \ldots, K_g\}$ has $K_g - \ell + 1$ users with a per-user storage of $\mathsf{m}_\ell = (M_\ell - M_{\ell-1})$.

• *File Splitting*: Next, each file F_n, is split into K_g non-overlapping fragments of size $(\alpha_1, \alpha_2, \ldots, \alpha_{K_g})B$ bits such that $\sum_{i=1}^{K_g} \alpha_i = 1$. The LHC scheme is based on the premise that the layer $\mathcal{L}_\ell$ is used to deliver α_ℓ fragment of the files requested by $K_g - \ell + 1$ users via multicast transmission. Resultantly, the ℓ^{th} user receives $\alpha_1 + \alpha_2 + \ldots + \alpha_\ell$ fraction of its requested file via ℓ multicasts. The remaining $\left(1 - \sum_{i=1}^{\ell} \alpha_i\right)$ fraction is delivered via unicast transmission.

The cache layering and file splitting strategies are shown in Fig. 2(a) for $K = 3$ users. The overall proposed LHC scheme using these ingredients is presented in Algorithm 1.

Algorithm 1 Layered Heterogeneous Caching

1: INITIALIZE: Split caches in $\mathcal{M}_g$ into K_g layers $\mathcal{L}_{1:K_g}$, such that $\mathsf{m}_\ell = M_\ell - M_{\ell-1}, \ell = 1, 2, \ldots, K_g$. Split each file into fragments $\alpha_1, \alpha_2, \ldots, \alpha_{K_g}$.
2: **for** each layer $\mathcal{L}_\ell$, with $\ell = 1, 2, \ldots, K_g$ **do**
3: CACHE STORAGE: Use centralized cache storage scheme from [1] for N files, $K_g - \ell + 1$ users, each with a cache storage of $\mathsf{m}_\ell B / \alpha_\ell$ bits.
At layer $\mathcal{L}_{K_g}$, store m_{K_g}/N fraction of the fragment α_K of each file which have not yet been stored in the cache of user K.
4: FILE DELIVERY:
(a) Deliver $\alpha_\ell B$ bits of requested files of users $\{\ell, \ell + 1, \ldots, K_g\}$ via multicast transmission.
(b) Deliver the remaining $\left(1 - \sum_{i=1}^{\ell} \alpha_i\right)$ bits of user ℓ's requested file via unicast.
(c) At the last layer $\mathcal{L}_{K_g}$, deliver the remaining $\left(\alpha_{K_g} - \mathsf{m}_{K_g}/N\right) B$ bits of user K_g's requested file via unicast.
5: **end for**

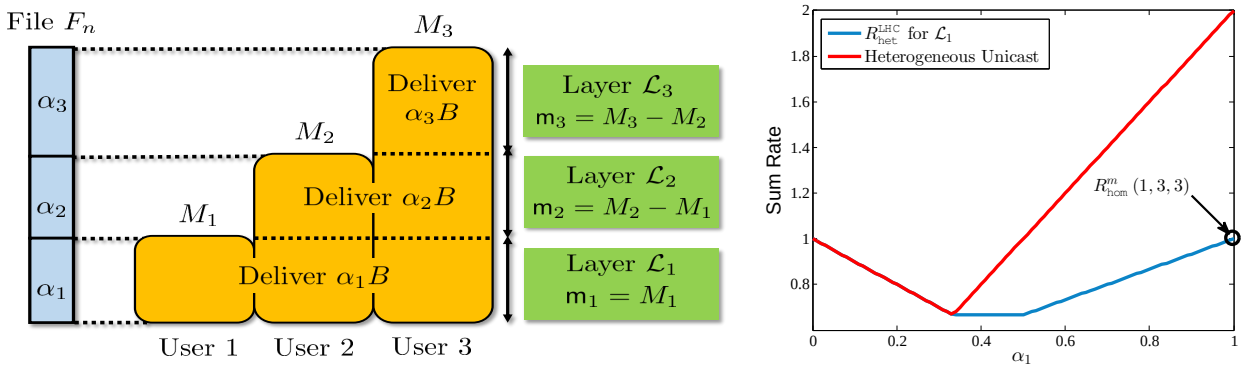
1) Achievable Rate of LHC Scheme: The achievable rate of the LHC scheme for the cache group $\mathcal{M}_g$ is the sum of the rates over the K_g layers. Focusing on the ℓ-th layer, note that the transmission has two components:

1. *Unicast* of $(1 - \sum_{i=1}^{\ell} \alpha_i)$ fraction of the file requested by the ℓ-th user.
2. *Multicast* of α_ℓ fraction of files requested by the set of $(K_g - \ell + 1)$ users, each user having a storge of $\mathsf{m}_\ell = M_\ell - M_{\ell-1}$.

We next separately analyze the unicast and multicast rates for the LHC scheme. To analyze the unicast rate, note that at layer $\mathcal{L}_\ell,\ \forall \ell \in \{1, 2, \ldots, K_g - 1\}$, the unicast rate for user ℓ is given by $1 - \sum_{i=1}^{\ell} \alpha_i = \sum_{i=\ell+1}^{K_g} \alpha_i$. For the final layer $\mathcal{L}_K$, the unicast rate of the K_g-th user is given by $\left(\alpha_{K_g} - \frac{\mathsf{m}_{K_g}}{N}\right)$. The unicast rate for all K_g layers is given as:

$$\text{Unicast Rate} = \sum_{i=2}^{K_g} (i-1)\alpha_i + \left(\alpha_{K_g} - \frac{\mathsf{m}_{K_g}}{N}\right). \tag{14}$$

For multicast delivery, the centralized caching scheme [1] is used in each layer $\mathcal{L}_\ell,\ \forall \ell \in \{1, 2, \ldots, K_g - 1\}$ to deliver α_ℓ fragment of each file. The following lemma (proved in the Appendix) gives an achievable *multicast* rate for each layer in the LHC scheme.



(a) (b)

Fig. 2. (a) Layered Heterogeneous Caching for $K = 3$ users; (b) Scaling of $R_{\text{het}}^{\text{LHC}}$ with α_1 for $N = K = 3$ and $M = 1$.

Lemma 1. *For any N files and K users, each with cache storage of MB bits, the achievable multicast rate for delivering $\alpha B \in (0, B]$ part of all requested files, is given by:*

$$R(\alpha, M, N, K) = \alpha R_{\text{hom}}^{m}\left(M/\alpha, N, K\right), \tag{15}$$

where $R_{\text{hom}}^{m}(M, N, K)$ is the achievable rate given in (8).

Using Lemma 1, the multicast rate for the ℓ-th layer is given by $\alpha_\ell R_{\text{hom}}^{m}\left(\frac{\mathsf{m}_\ell}{\alpha_\ell}, N, K_g - \ell + 1\right)$. Thus, we have:

$$\text{Multicast Rate} = \sum_{\ell=1}^{K_g - 1} \alpha_\ell R_{\text{hom}}^{m}\left(\frac{\mathsf{m}_\ell}{\alpha_\ell}, N, K_g - \ell + 1\right). \tag{16}$$

Thus, for a given $\vec{\alpha}_g = \{\alpha_1, \alpha_2 \ldots, \alpha_{K_g}\}$, the achievable rate of LHC for the cache set $\mathcal{M}_g$ is given by:

$$R_{\text{het}}^{\text{LHC}}\left(\mathcal{M}_g, N, K_g, \vec{\alpha}_g\right) = \underbrace{\sum_{i=2}^{K_g}(i-1)\alpha_i + \left(\alpha_{K_g} - \frac{\mathsf{m}_{K_g}}{N}\right)}_{\text{Unicast Rate}}$$

$$+ \underbrace{\sum_{\ell=1}^{K_g - 1} \alpha_\ell R_{\text{hom}}^{m}\left(\frac{\mathsf{m}_\ell}{\alpha_\ell}, N, K_g - \ell + 1\right)}_{\text{Multicast Rate}}. \tag{17}$$

The achievable rate in (17) can be minimized by choosing the optimal $\vec{\alpha}_g^*$ as follows:

$$\vec{\alpha}_g^* = \arg\min_{\vec{\alpha}_g} \; R_{\text{het}}^{\text{LHC}} \quad \text{s.t.} \sum_{i=1}^{K_g} \alpha_i = 1, \; 0 \le \alpha \le 1. \tag{18}$$

The solution yields the best achievable LHC rate $R_{\text{het}}^{\text{LHC}}\left(\mathcal{M}_g, N, K_g, \vec{\alpha}_g^*\right)$ for the cache set $\mathcal{M}_g$. The rate, when evaluated over all subsets $\mathcal{M}_g$ such that $g \in \mathcal{G}^{\text{opt}}$ i.e., over the optimal partitioning of $\mathcal{M}$, yields the upper bound on the optimal rate for heterogeneous caching in (12).

Remark 1 (*Optimal File Splitting*)**.** Fig. 2(b) illustrates the intuition behind the choice of optimal α for any given layer in the LHC. Consider the system in Fig. 2(a) for $N = 3$ files and a storage of $\mathsf{m}_1 = 1$ in layer $\mathcal{L}_1$. Fig. 2(b) shows the heterogeneous unicast and LHC rates as a function of the split α_1. The rates are minimized at $\alpha_1 = 1/3$. However, a larger fraction, $\alpha_1 = 0.5$, can be delivered in this layer without increasing the sum-rate when LHC is used. This ensures maximum utilization of storage at layer $\mathcal{L}_1$. For a homogeneous system, with $M_1{=}M_2{=}M_3{=}1$, however, $\alpha_1 = 1$ is optimal since the entire file has to be delivered in layer $\mathcal{L}_1$. $\diamond$

C. Information Theoretic Lower Bound

The next theorem presents a cut-set based lower bound on the optimal rate for the heterogeneous caching problem.

Theorem 2. *For any N files and K users with heterogeneous cache storage $\mathcal{M}$, we have*

$$R_{\text{het}}^{*}(\mathcal{M}, N, K) \ge \max_{\substack{s \in 1,\ldots,\min\{N,K\} \\ M_i \in \mathcal{M}, \forall i \in [s]}} \left(\frac{N - \sum_{i=1}^{s} M_i}{\lceil N/s \rceil}\right). \tag{19}$$

For systems with 2 and 3 levels of heterogeneity, the LHC scheme is order-optimal i.e., the upper and lower bounds are within constant multiplicative gaps of 19, 28 respectively. The proofs of the lower bound and order-optimality are omitted due to lack of space and detailed in [13].

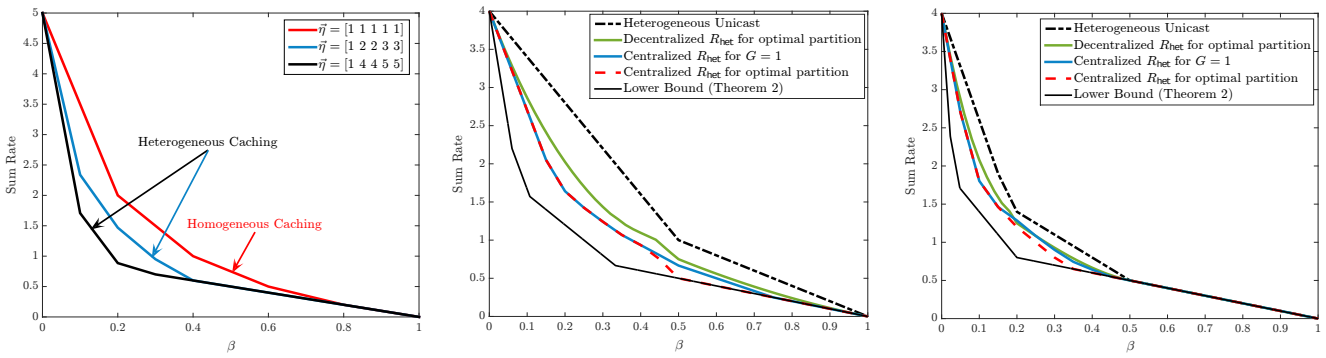
V. Illustration of Results

The heterogeneity of $\mathcal{M}$ is characterized by two parameters namely $\vec{\eta} = [\eta_1, \eta_2, \ldots, \eta_K]$ such that $\eta_i = M_i/M_1, \forall i \in [K]$, and $\beta = M_1/N$. To characterize the storage-rate tradeoff, the sum-rate of the system is plotted against increasing β for a given heterogeneity $\vec{\eta}$. Figure 3(a), shows that the achievable rate of the LHC scheme with $N = K = 5$, decreases with increasing heterogeneity for the optimal partitioning of caches due to increase is system cache storage.

Homogeneous Caching: LHC is modeled on cache layering and file-splitting with each layer operating independently. Thus LHC, when applied to a system with homogeneous caching, uses only one layer with the entire cache memory and all users, to deliver requested files yielding the multicast rate in [1].

Decentralized Storage: In LHC, within each layer, the cache storage phase is centralized. However, due to the generality of the layering approach, a decentralized cache storage scheme based on random caching [2], can be used. In this case, each user k can randomly store any M_k/N bits in its cache. The server then divides the cache content of each user ℓ in layer $\mathcal{L}_\ell$ into ℓ non-overlapping fragments of size m_ℓ. The server uses the multicast delivery scheme in [2] for each layer and optimizes over $\vec{\alpha}$ to determine which fraction of files to deliver. The achievable rate of LHC in (17) is then based on the rate of decentralized delivery [2] in each layer.

Figures 3(b)-3(c) show the achievable rate for heterogeneous unicast along with centralized and decentralized LHC schemes



(a) (b) (c)

Fig. 3. $R_{\text{het}}(\mathcal{M})$ trade-off for (a) $N = K = 5$ (b) $N = 10, K = 4$, with $\vec{\eta} = [1\ 1\ 2\ 2]$; (c) $N = 10, K = 4$, with $\vec{\eta} = [1\ 2\ 5\ 6]$.

for $N = 10$ and $K = 4$. Figure 3(c) shows that, as heterogeneity increases the LHC rate approaches the unicast rate. This is due to the fact that with increase in heterogeneity, layers with more users i.e., with more multicasting opportunities, have low cache memory leading to lesser multicast gains. Therefore the first term in LHC rate in (17) dominates. Furthermore, decentralized caching faces a rate loss due random storage which also decreases with heterogeneity due to lack of multicast gains. Figures 3(b)-3(c) also show that the rate of Theorem 1 converges to the lower bound of Theorem 2 for large β.

Cache Partitioning: For low heterogeneity, using a single partition $\{1, 2, \ldots, K\}$ with $G = 1$, provides similar performance compared to optimal partitioning with $G \geq 1$. However with increasing heterogeneity, partitioning better captures the disparity in cache sizes to provide maximal multicasting gains.

VI. Conclusions and Future Directions

In this paper, we introduced the heterogeneous caching problem and presented a new achievable scheme based on set partitioning and cache layering. The proposed Layered Heterogeneous Caching scheme utilizes the system storage maximally and delivers content via a combination of multicast and unicast delivery. We also presented a lower bound on the optimal rate. We showed that as heterogeneity in storage increases, the multicasting gain reduces and usefulness of optimal user-partitioning increases. Based on the proposed framework, an interesting direction of future work is to design a low-complexity (polynomial time) algorithm for finding $\mathcal{G}^{opt}$. Furthermore, showing the order optimality of LHC for K levels of heterogeneity remains an open problem.

Appendix

Each user has a cache storage of MB bits in which they can each store $\frac{M}{N}$ fragments of each of the N files. Now assume each file is of size $B^{'} = \alpha B$ bits. We let each user store $\frac{M}{\alpha N}$ fragment of each file. Now consider the case where $\frac{M}{\alpha} = \frac{Nt}{K}, t \in \{1, \ldots, K\}$. In this case we have $t \triangleq \frac{KM}{\alpha N}$. Following the storage and multicast delivery scheme in [1], each file of size $B^{'}$ bits is then divided into $\binom{K}{t}$ sub-files of size $B^{'}/\binom{K}{t}$ bits. Each user caches a total of $N\binom{K-1}{t-1}$ of these sub-files i.e., each user caches a total of

$$N\binom{K-1}{t-1}\frac{B^{'}}{\binom{K}{t}} = \alpha B\frac{Nt}{K} = MB \quad \text{bits}, \tag{20}$$

which satisfies the cache storage constraint. Further, in the delivery phase, $\binom{K}{t+1}$ transmissions, each of size $B^{'}/\binom{K}{t}$ bits are made. Thus the achievable rate R is given by:

$$RB = \binom{K}{t+1}\frac{B^{'}}{\binom{K}{t}} = \alpha B\frac{K-t}{t+1}$$

$$\Rightarrow \quad R = \alpha\frac{K\left(1-\frac{M}{\alpha N}\right)}{1+\frac{KM}{\alpha N}} = \alpha R^{m}_{\text{hom}}\left(\frac{M}{\alpha}, N, K\right). \tag{21}$$

For any other $M/\alpha \in (0, N]$, cache splitting and time-sharing [1] can achieve the rate $\alpha R^{m}_{\text{hom}}\left(\frac{M}{\alpha}, N, K\right)$, where R^{m}_{hom} is given in (8) for $t = \left\lfloor \frac{KM}{\alpha N} \right\rfloor$.

References

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

[2] ——, "Decentralized coded caching attains order-optimal memory-rate tradeoff," *IEEE/ACM Transactions Networking*, vol. 23, no. 4, pp. 1029–1040, Aug. 2015.

[3] U. Niesen and M. A. Maddah-Ali, "Coded Caching with Nonuniform Demands," in *Proc. IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, April 2014, pp. 221–226.

[4] A. Sengupta, R. Tandon, and T. C. Clancy, "Improved Approximation of Storage-Rate Tradeoff for Caching via New Outer Bounds," in *Proc. IEEE International Symposium on Information Theory (ISIT)*, June 2015.

[5] N. Ajaykrishnan, N. S. Prem, V. M. Prabhakaran, and R. Vaze, "Critical Database Size for Effective Caching," *arXiv:1501.02549*, 2015. [Online]. Available: http://arxiv.org/abs/1501.02549

[6] H. Ghasemi and A. Ramamoorthy, "Improved lower bounds for coded caching," in *Proc. IEEE International Symposium on Information Theory (ISIT)*, June 2015, pp. 1696–1700.

[7] M. Ji, A. M. Tulino, J. Llorca, and G. Caire, "Order Optimal Coded Caching-Aided Multicast under Zipf Demand Distributions," in *The Eleventh International Symposium on Wireless Communication Systems (ISWCS)*, 2014.

[8] N. Karamchandani, U. Niesen, M. Maddah-Ali, and S. Diggavi, "Hierarchical Coded Caching," in *Proc. IEEE International Symposium on Information Theory (ISIT)*, June 2014, pp. 2142–2146.

[9] A. Sengupta, R. Tandon, and T. C. Clancy, "Fundamental Limits of Caching with Secure Delivery," *IEEE Transactions on Information Forensics and Security*, vol. 10, pp. 355–370, Feb 2015.

[10] M. Ji, G. Caire, and A. F. Molisch, "Fundamental limits of caching in wireless D2D networks," *arXiv:1405.5336*, 2014. [Online]. Available: http://arxiv.org/abs/1405.5336

[11] A. Sengupta and R. Tandon, "Beyond Cut-Set Bounds-The Approximate Capacity of D2D Networks," in *Proc.Information Theory and Applications (ITA)*, February 2015.

[12] H. S. Wilf, *Generatingfunctionology*. Natick, MA, USA: A. K. Peters, Ltd., 2006.

[13] A. Sengupta, R. Tandon, and T. C. Clancy, "Layered caching for heterogeneous storage." [Online]. Available: https://filebox.ece.vt.edu/~aviksg/LHC.pdf