Audience-Retention-Rate-Aware Caching and Coded Video Delivery with Asynchronous Demands

Qianqian Yang, Mohammad Mohammadi Amiri, and Deniz Gündüz

***Abstract*—Most of the current literature on coded caching focus on a static scenario, in which a fixed number of users synchronously place their requests from a content library, and the performance is measured in terms of the latency in satisfying all of these requests. In practice, however, users start watching an online video content asynchronously over time, and often abort watching a video before it is completed. The latter behaviour is captured by the notion of *audience retention rate*, which measures the portion of a video content watched on average. In order to bring coded caching one step closer to practice, asynchronous user demands are considered in this paper, by allowing user demands to arrive randomly over time, and both the popularity of video files, and the audience retention rates are taken into account. A decentralized *partial coded delivery* (PCD) scheme is proposed, and two cache allocation schemes are employed; namely *homogeneous cache allocation* (HoCA) and *heterogeneous cache allocation* (HeCA), which allocate users' caches among different chunks of the video files in the library. Numerical results validate that the proposed PCD scheme, either with HoCA or HeCA, outperforms conventional uncoded caching as well as the state-of-the-art decentralized caching schemes, which consider only the file popularities, and are designed for synchronous demand arrivals. An information-theoretical lower bound on the average delivery rate is also presented.**

I. Introduction

The ever-increasing demand for video content has been the main driver of the recent explosive growth of wireless data traffic. A key feature of video data is that a small portion of highly popular contents dominate the traffic [2]. This led to the idea of prefetching popular contents over off-peak traffic periods, or at better channel conditions, and storing them at the network edge [3], [4], or even directly at user devices [5], [6], referred to as *proactive caching*. Proactive caching can reduce both the traffic load on the backhaul links and the latency; and it has become viable thanks to the decreasing cost of memory (see [3], [5]–[7], and references therein).

In proactive caching, users' caches are filled without the knowledge of future user demands, referred to as the *placement phase*. Users' demands are revealed during the peak traffic period, and are satisfied simultaneously over the *delivery phase*. Traditional uncoded caching schemes adopt orthogonal unicast transmissions, so the caching gain is limited by the available cache memory. On the other hand, *coded caching* [6] exploits the cache resources across the network by jointly optimizing the two phases in order to create and exploit coded multicasting opportunities, even among distinct user requests. It is shown in [6] that coded caching provides a global caching gain, which depends on the total cache capacity in the network. Coded caching and delivery has ignited intense research activities in recent years [8]–[15].

There are two limitations of the current literature that we address in this paper: The first is the *synchronous demand assumption*, that is, all the requests arrive simultaneously at the beginning of the delivery phase. The other limitation is that the *users request entire files*. However, in practice, users rarely request and watch an entire video content, and different user requests may arrive at different time instants, and each user may abort watching a certain video content after a random duration. A recent report [2] suggests from a trace of 7000 Youtube videos that, users on average watch 60% of their requested files, and the number of views varies over different videos as well as different parts of each video. This phenomena is captured by the *audience retention rate*, used by online video platforms, such as Youtube and Netflix, to model the popularity of different parts of a content. It is provided to content generators to better understand user engagement. For efficient caching and delivery, this nonuniform viewing behaviour calls for *partial caching*, where only the most viewed portion of each video file is cached.

Several papers have considered dynamic models with coded caching. A slotted time model is considered in [13], where the library of popular files change dynamically from one time slot to the next in a Markovian fashion. However, users are still assumed to place their requests simultaneously at each time slot for complete files in the library at that time slot. In [16] a delay-sensitive demand model is considered, where each user has a different deadline by which her demand must be satisfied. This model is extended in [17] by assuming that the demands arrive at different times, which, to the best of our knowledge, is the only prior work considering asynchronous demand arrivals. However, the two cases considered in [17] may not reflect the real behaviour in video steaming applications; in the offline scenario, the authors assume that the arrival times are known at the beginning of the delivery phase; on the other hand, in the online scenario, the total completion time is minimized without the knowledge of the demand arrival times assuming

The authors are with Imperial College London, London SW7 2AZ, U.K. (e-mail: q.yang14@imperial.ac.uk; m.mohammadi-amiri15@imperial.ac.uk; d.gunduz@imperial.ac.uk).

This paper was presented in part at the IEEE International Conference on Communications, Workshop on Advanced Caching for Wireless Networks, Paris, France, May 2017 [1].

This work received support from EC H2020-MSCA-ITN-2015 project SCAVENGE under grant number 675891, and from the European Research Council project BEACON under grant number 677854.

that each user has a relatively loose deadline on receiving its requested file. In [14] users are allowed to request the same file at different resolutions. However, this model is still limited to complete requests, albeit at different qualities, for each user. Audience retention rate aware partial caching is shown to improve the performance of uncoded caching in [18].

Here, we investigate coded caching of video files taking into account the audience retention rate for each video. We consider that each video file consists of equal-length chunks, and the audience retention rate of each chunk is the fraction of users watching this chunk among total views of the corresponding video. Also, in contrast to the literature on coded caching, where users are assumed to reveal their demands simultaneously, we consider a more realistic asynchronous and random demand model, where users dynamically join the delivery phase over time, and leave the system after watching a random number of video chunks. We propose a novel decentralized caching and coded delivery scheme, referred to as the *partial coded delivery* (PCD). We derive a closed-form expression for the achievable average delivery rate over all possible demand combinations, taking both the asynchronous demand arrivals and audience retention rate into account. Thanks to this closed-form expression for the long-term achievable average delivery rate, we then employ a cache allocation scheme, referred to as the *heterogeneous cache allocation* (HeCA), in order to minimize the average delivery rate. We also propose an alternative simplified cache allocation scheme, called *homogeneous cache allocation* (HoCA), where the most popular chunks are cached by each user. Finally we derive an information-theoretic lower bound on the achievable average delivery rate. We remark here that the coded caching problem with different file popularities, studied in [11], [19], is a special case of the problem considered here, obtained by setting the audience retention rates of all the chunks to one, and assuming all the demands arrive simultaneously. Numerical results indicate that the proposed audience retention rate aware partial coded delivery scheme achieves a better delivery rate than both uncoded delivery and the scheme proposed in [11] adapted to the current setting.

The rest of this paper is organized as follows. The system model is introduced in Section II. In Section III, we introduce the proposed PCD scheme, and analyze its performance in terms of the average delivery rate. We present a lower bound in Section IV. Numerical results are presented in Section V. Finally, we conclude the paper in Section VI, followed by the Appendices.

Notations: We denote the set of t-bit binary sequences by $[2^t]$, and all binary sequences by $[2^*]$. The set of integers $\{i,\dots,j\}$, $i \leq j$, is denoted by $[i:j]$, while, $\{1,\dots,j\}$ is denoted by $[j]$. For sets $\mathcal{A}$ and $\mathcal{B}$, $\mathcal{A}\backslash\mathcal{B} \triangleq \{x : x \in \mathcal{A}, x \notin \mathcal{B}\}$, and $|\mathcal{A}|$ denotes the cardinality of $\mathcal{A}$. Notation $\overline{\oplus}$ represents the bitwise XOR operation, where the arguments are zero-padded to have equal length. For two positive integers i, j, $i \leq j$, $K_{i:j}$ denotes $(K_i,\dots,K_j)$; while $K_{[j]}$ denotes $(K_1,\dots,K_j)$. For event E, $\mathbb{1}\{E\} = 1$, if E is true; and $\mathbb{1}\{E\} = 0$, otherwise. $\binom{j}{i}$ represents the binomial coefficient, if $j \geq i$; and $\binom{j}{i} = 0$, otherwise. $\mathbb{R}$ and $\mathbb{N}$ denote the sets of real numbers and positive integers, respectively.

II. System Model

We consider a server with a library of N video files, denoted by $\mathcal{F} = \{W_1, ..., W_N\}$. We assume, for simplicity, that all the files have the same size of F bits. Each file consists of B chunks of equal size, i.e., F/B bits each. We denote by W_{ij} the jth chunk of file W_i.

In the placement phase, each user pre-fetches data from the server to fill its cache of size MF bits. We consider a dynamic delivery phase; that is, users arrive randomly, request a random video from the library, watch a random number of chunks of that video, and leave the system. Active users at any time instant are connected to the server through an error-free shared link.

We consider a slotted time model, where the beginning of the delivery phase is marked as $t = 0$, and the unit time interval $(t-1, t]$ is referred to as time slot t, $t \in \mathbb{N}$. We assume that a user consumes exactly one chunk of a video file in one time slot. We denote the number of new demands that arrive during time slot t as a_t, where a_t is independently and identically distributed (i.i.d.) according to P_A over a finite set $\mathcal{A} = \{0, 1, ..., A_{\max}\}$; that is, only a limited number of new users can be admitted at each time slot. Each demand corresponds to a file from $\mathcal{F}$, i.i.d. according the *popularity distribution* of the library $\mathbf{p} \triangleq (p_1, ..., p_N)$.

Unlike the current literature, we do not necessarily deliver the requested contents in their entirety, as users often quit watching a video file before completion. Therefore, in our model, users are initially delivered only the first chunks of their desired video files. Their demands for subsequent chunks are revealed only after receiving the previous ones. Specifically, the first chunks of the a_t demands that arrived in slot t are delivered during slot $t+1$, and then the corresponding a_t users decide to continue watching or not after having received the first chunks. Those who have decided to continue watching are served the second chunks of their requested files during slot $t+2$. In the same manner, having received jth chunks during slot $t+j$, the users who continue watching are delivered the $(j+1)$th chunks during slot $t+j+1$, for $j \in [B-1]$. We note that the first chunks of the requested files are always delivered, and at any time slot t, the server may be serving demands that have arrived at time slots $t-B, t-B+1, \dots, t-1$.

To model this, we employ the notion of *audience retention rate*, defined as the fraction of users that request chunk W_{ij} among all the users that have requested W_i, denoted by p_{ij}, for $i \in [N]$ and $j \in [B]$ [18]. Alternatively, we can regard p_{ij} as the probability that a user who requested video W_i will watch the jth chunk[1]. Accordingly, p_{ij} is non-increasing in j, i.e., $1 = p_{i1} \geq p_{i2} \geq \cdots \geq p_{iB}$, which characterizes a realistic viewing model that users start watching videos from the beginning and abort after watching a random number of chunks in order. We let $\mathbf{P} = \{p_{ij}, i \in [N], j \in [B]\}$ denote the *retention rate matrix* for all the chunks in the library, which is time-invariant and identical for all the users. We refer to $p_i p_{ij}$ as the popularity of chunk W_{ij} in the sense that it denotes the

[1]Here we assume that a user cannot skip chunk W_{ij} for some $j \in [B]$, and request a later chunk k, for $k > j$. Once a user does not request chunk W_{ij}, it leaves the system, and does not receive any further content.

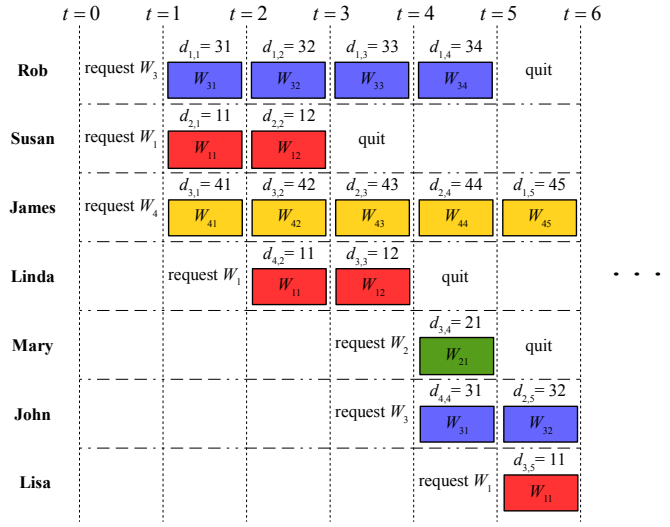

Fig. 1. Illustration of the demand arrivals for an asynchronous caching system with $N \geq 4$ files and $A_{\max} \geq 3$ for time slots $t = 1$ to 6 of the delivery phase. In the caching setting under consideration, we have $a_1 = 3$, $a_2 = 1$, $a_3 = 0$, $a_4 = 2$, $a_5 = 1$ demands, and $K^{(1)} = 3$, $K^{(2)} = 4$, $K^{(3)} = 3$, $K^{(4)} = 4$, $K^{(5)} = 3$ users served at each time slot.

probability that chunk W_{ij} will be requested by a user that joins the system.

Remark 1. *We remark that the coded caching model considered in this paper does not specify the chunk size. However, if the chunk size is too small, the computational complexity may be prohibitive. Moreover, we expect that there will not be a significant difference between the chunks in terms of the audience retention rate if they become too small. Note also that the length of a time slot is set to be the duration that a user watches one chunk without pausing. And in our system model, users start watching their requests with a delay of at most one time slot. Hence, the chunk size is constrained by the delay tolerance, the frame rate of the video, as well as the display settings of user devices [20].*

During the placement phase, each user fills its cache as an arbitrary function of the library $\mathcal{F}$, the file popularity vector $\mathbf{p}$, and the retention rate matrix $\mathbf{P}$, subject to its cache capacity of MF bits. We emphasize that the knowledge of the future requests is not available during the placement phase, and only during the placement phase the contents in the caches are updated. We also note that the placement phase is performed in a decentralized manner; that is, coordination among the users during this phase is not possible since the server does not know when a user is going to make a request in advance.

The delivery phase begins once the users start requesting files, and as described above, is performed over many time slots. At each slot t, the server serves all the active users in the system, those from slot $t-1$ that continue watching their requested contents, as well as the new arrivals. Denote by $K^{(t)}$ the total number of active users to be served at time slot t, and $K_j^{(t)}$ the number of users among the $K^{(t)}$ active users requesting their jth chunks. All the active users are re-indexed at the beginning of slot t as $[K^{(t)}]$ in a way that users $\sum_{h=1}^{j-1} K_h^{(t)} + 1$ to $\sum_{h=1}^{j} K_h^{(t)}$ are requesting their jth chunks, $j = 1, ..., B$. The cache content of the kth user, for $k \in [K^{(t)}]$, is denoted by $Z_k^{(t)}$. Let $d_{k,t}$ denote the index of the chunk requested by user k, which needs to be delivered at slot t, $d_{k,t} \in \{ij : i \in [N], j \in [B]\}$. We remark that if user k has joined the system at slot t', then $d_{k,t} \in \{ij : i \in [N], j = t - t'\}$, i.e., at slot t, the $(t-t')$th chunk of her request will be delivered to user k. Let $\mathbf{d}_t \triangleq (d_{1,t}, ..., d_{K^{(t)},t})$ denote the demand vector at slot t, and $\mathcal{D}_t \triangleq \{W_{d_{1,t}}, ..., W_{d_{K^{(t)},t}}\}$ denote the set of requested chunks. Then, to satisfy all these requests at slot t, the server sends a message of length $R_{\mathbf{d}_t} F/B$ bits over the shared link, which is a function of the library $\mathcal{F}$, the demand vector $\mathbf{d}_t$, and the cache contents of the active users $Z_1^{(t)}$, ..., $Z_{K^{(t)}}^{(t)}$. User k recovers chunk $W_{d_{k,t}}$ at the end of slot t from the transmitted message and the contents in her local cache. We are interested in the long-term average delivery rate

$$R \triangleq \lim_{T \to \infty} \mathbb{E}\left[\frac{1}{T} \sum\nolimits_{t=1}^{T} R_{\mathbf{d}_t}\right], \tag{1}$$

where the expectation is taken over all demand realizations $\mathbf{d}_t$ distributed according to P_A, $\mathbf{p}$, and $\mathbf{P}$.

It is easy to see that $\mathbf{d}_t$ is a Markov chain, and since the cached contents of active users are constant throughout the delivery phase, $R_{\mathbf{d}_t}$ at time t depends only on the current state $\mathbf{d}_t$; therefore, the long-term average rate in (1) can be obtained by simply evaluating $\mathbb{E}[\sum_{t=1}^{T} R_{\mathbf{d}_t}]$ in the steady state demand distribution, which depends on P_A, $\mathbf{p}$, and $\mathbf{P}$.

Definition 1. *A cache capacity-average rate pair (M, R) is achievable for the caching system described above, if there exists a caching and delivery scheme with cache capacity M at each user and average rate R such that for any demand realization $\mathbf{d}_t$, $\forall t$,*

$$\lim_{F/B \to \infty} \Pr\left\{\bigcup\nolimits_k \left\{\hat{W}_{d_{t,k}} \neq W_{d_{t,k}}\right\}\right\} = 0, \tag{2}$$

where $\hat{W}_{d_{k,t}}$ denotes the reconstruction of $W_{d_{k,t}}$ at user k at the end of time slot t.

We define $R^*(M) \triangleq \inf\{R : (M, R) \text{ is achievable}\}$ to express the tradeoff between the cache capacity and the average delivery rate. The goal in this paper is to characterize this trade-off.

Remark 2. *The delivery rate $R_{\mathbf{d}_t}$ as defined above (following [8]) refers to the total number of bits (normalized by F/B) that must be delivered in order to satisfy all user demands in time slot t. Therefore, it can be considered as a measure of latency, rather than the more classical communication rate concept. In our setting, however, we consider a slotted system; hence, the duration of each time slot is considered fixed according to the display duration of one chunk of a video file. Accordingly, $R_{\mathbf{d}_t}$ can be considered as a measure of the bandwidth/capacity required to satisfy all user demands within a time slot duration to guarantee the streaming of video files without stalling.*

III. Partial Caching and Coded Delivery(PCD)

In this session, we present our cache placement scheme, followed by the proposed coded delivery scheme, referred to

Algorithm 1 Random Delivery

for $W_{ij} \in \mathcal{D}_t$ **do**
    Server sends enough random linear combinations of the bits of file W_{ij} to enable the users demanding it to decode it.
end for

Algorithm 2 Delivery scheme at time slot t based on [8, Algorithm 1]

Delivering the missing bits that are in the cache of any subset of users in $\mathcal{K}_t$:
for $z = 1, ..., K^{(t)}$ **do**
    for $\mathcal{P} \subset [K^{(t)}]$: $|\mathcal{P}| = z$ **do**
        Send $\overline{\bigoplus}_{k\in\mathcal{P}} W^t_{d_{k,t},\mathcal{P}\setminus\{k\}}$
    end for
end for

as the *partial coded delivery* (PCD), along with an example. We derive the average delivery rate achieved by PCD, based on which the cache allocation is optimized. We remark that the number of bits and the delivery rate mentioned in the sequel are both normalized by F/B.

A. Placement Phase

During the placement phase, each user selects an independent random subset of $q_{ij}F/B$ bits from W_{ij} to fill its cache, where $0 \leq q_{ij} \leq 1$, such that $\sum_{i=1}^{N}\sum_{j=1}^{B} q_{ij} = MB$, which, for large F, satisfies the cache capacity constraint with high probability. We refer to $\mathbf{Q} = \{q_{ij}, i \in [N], j \in [B]\}$ as the *cache content distribution*, which will be optimized in order to minimize the average delivery rate. The optimization of $\mathbf{Q}$ is studied in Section III-E.

B. Delivery Phase

As described in the system model, the delivery phase is performed over different time slots, according to the current demand configuration specified by $\mathbf{d}_t$ during each time slot t, and cache contents $Z^{(t)}_1, ..., Z^{(t)}_{K^{(t)}}$, where $K^{(t)}$ denotes the number of active users at slot t. We emphasize that users' requests for the jth chunks are revealed only after they receive the first $j-1$ chunks, and in the delivery phase a user is not served a chunk before requesting it. This is because the user's caches are filled in advance during the placement phase, and proactively delivering a new chunk would require removing contents from the caches. While this may potentially provide gains, we leave this as future work as dynamically adjusting cache contents while making decisions for proactive delivery will significantly change the problem formulation and the optimization techniques. For $\mathcal{S} \subset [K^{(t)}]$, we denote by $W^t_{ij,\mathcal{S}}$ the bits of chunk W_{ij} that are exclusively cached by the users in $\mathcal{S}$ (i.e., not cached by any of the users in $[K^{(t)}] \setminus \mathcal{S}$). We note that $W^t_{ij,\mathcal{S}}$ is not necessarily the same as $W^{t'}_{ij,\mathcal{S}}$ for $t \neq t'$, $t, t' \in \mathbb{N}$, since a different set of users may be active at each time slot; and thus, $\mathcal{S}$ may refer to a different subset of users at different time slots.

Algorithm 3 PCD scheme at time slot t

PART 1: Delivering the missing bits that are not in the cache of any user in $[K^{(t)}]$:
for $W_{ij} \in \mathcal{D}_t$ **do**
    Send $W^t_{ij,\emptyset}$
end for
PART 2: Delivering the missing bits that are in the cache of only one user in $[K^{(t)}]$; the one among **PART 2.1** and **PART 2.2** that requires a smaller delivery rate is executed:
PART 2.1:
for $\mathcal{P} \subset [K^{(t)}]$: $|\mathcal{P}| = 2$ **do**
    Send $\overline{\bigoplus}_{k\in\mathcal{P}} W^t_{d_{k,t},\mathcal{P}\setminus\{k\}}$
end for
PART 2.2:
for $W_{ij} \in \mathcal{D}_t$ **do**
    Send $\bigcup_{k=1}^{K^{(t)}-1} W^t_{ij,\{k\}} \overline{\oplus} W^t_{ij,\{k+1\}}$
end for
PART 3: Delivering the missing bits that are in the cache of more than one user in $[K^{(t)}]$:
for $\mathcal{P} \subset [K^{(t)}]$: $|\mathcal{P}| > 2$ **do**
    Send $\overline{\bigoplus}_{k\in\mathcal{P}} W^t_{d_{k,t},\mathcal{P}\setminus\{k\}}$
end for

In Algorithm 1, we present the *Random Delivery* (RAN) scheme, which simply delivers random linear combinations of the bits of a chunk until it is decoded by the requesting user. This scheme has been considered in [8], [12], [15] as an alternative delivery procedure although it is known to perform poorly in general compared to coded delivery.

The second delivery scheme considered here is presented in Algorithm 2. We will refer to it as the MAN scheme as it is based on [8, Algorithm 1]. We remark that, here we use operation $\overline{\bigoplus}$ instead of the $\bigoplus$ in [8, Algorithm 1]. Recall that $\overline{\oplus}$ represents the bitwise XOR operation, in which the arguments are zero-padded to have equal length. For each z value, $z \in [K^{(t)}]$, $\binom{K^{(t)}}{z}$ coded contents are delivered, each of which is targeted to a distinct set of z users denoted by $\mathcal{P}$. With each coded content, targeted to z users, each of them obtains its missing bits cached by other $z-1$ users. Accordingly, after performing the delivery phase presented in Algorithm 2, user k recovers $\bigcup_{\mathcal{P}\subset[K^{(t)}]:k\in\mathcal{P}} W^t_{d_{k,t},\mathcal{P}\setminus\{k\}}$, which together with its cache content $Z^{(t)}_k = \bigcup_{\mathcal{P}\subset[K^{(t)}]:k\in\mathcal{P}} W^t_{d_{k,t},\mathcal{P}}$ enable user k to recover $W^t_{d_{k,t}}$, for $k \in [K^{(t)}]$.

We note that the MAN scheme does not take into account the popularity of the files or chunks, and delivers the same number of coded contents for any demand vector $\mathbf{d}_t$. With the PCD scheme, we aim to reduce the repetition in the contents sent by MAN when the demands are not distinct. More specifically, the number of delivered contents depends on the demand vector $\mathbf{d}_t$ for $z = 1$ and $z = 2$ as shown in the sequel.

The novel PCD scheme, first introduced in [1] for synchronous user demands, is presented in Algorithm 3. Here we will optimize and analyze its performance for asynchronous user demands. PART 1 sends the missing bits that are not cached by any user, which correspond to the content sent by

the MAN scheme for $z = 1$. However, unlike MAN, it avoids repeating the bits of the chunks requested by multiple users. PART 2 sends the missing bits that are cached by only one user, where PART 2.1 is the same with the MAN scheme for $z = 2$. PART 2.2 presents an alternative delivery method with a smaller delivery rate than PART 2.1 when the same chunk is requested by a sufficient number of users. We note here that Part 1 and Part 2 can be applied to the conventional caching setting considered in [8] and improve the MAN scheme as long as the average performance is considered. We also note that the idea used in Part 2.2 can potentially be extended to $z > 2$. However, we have not been able to find the scheme that leads to a closed-form expression for the average delivery rate. We leave this extension as future work. PART 3 is the same with the MAN scheme for $z > 2$. Hence, PCD outperforms MAN, where the improvement comes from sending the missing bits that are cache by none or only one of the users more efficiently.

Here we further investigate the difference between PART 2.1 and PART 2.2. For F/B large enough, the expected length of $W^t_{ij,k}$ is given by $(q_{ij})\left(1-q_{ij}\right)^{K^{(t)}-1} F/B$ bits, for any $k \in [K^{(t)}]$. We define $T_{ij,1} \triangleq \left|W^t_{ij,k}\right|$, for $ij \in \mathbf{d}_t$, and $\forall k \in [K^{(t)}]$. For the case when q_{ij} is identical for any i and j, we note that $T_{ij,1}$, $\forall ij \in \mathbf{d}_t$, are also identical, and we denote $T_1 = T_{ij,1}$. The numbers of bits sent by PART 2.1 and PART 2.2 are $\binom{K^{(t)}}{2} T_1$ and $|\mathcal{D}_t|(K^{(t)}-1)T_1$, respectively, where we remind that $K^{(t)}$ is the set of users to be served in time slot t, and $\mathcal{D}_t$ is the set of the requested chunks at time slot t. Note that $|\mathcal{D}_t|(K^{(t)}-1)T_1 < \binom{K^{(t)}}{2} T_1$ if $|\mathcal{D}_t| < K^{(t)}/2$, which implies that PART 2.2 outperforms PART 2.1 when the number of distinct demands is less than half of the number of users. This is likely to happen with nonuniform popularity and larger number of users than files. Consider a simple example with 5 users in the system, and assume that there are 4 chunks with popularities $0.7, 0.15, 0.1, 0.05$, respectively. The probability that these 5 users request no more than 2 distinct files is 0.68. For the general case, the difference between PART 2.1 and Part 2.2 is analysed in Section III-D in terms of the average delivery rate.

Remark 3. *We remark that the PCD scheme can be applied to the conventional caching setting as the one studied in [8], regardless of audience retention or asynchronous demand arrivals. Part 1 and Part 2 of Algorithm 3 are beneficial as long as the average performance is considered instead of worst-case. We also note that the coded delivery scheme in [15, Algorithm 1] in general can achieve a lower delivery rate than the above schemes for the same demand combination. However, the average delivery rate of the scheme in [15, Algorithm 1] does not lend itself to a closed-form expression; and therefore, it is challenging to optimize cache allocation. The proposed PCD scheme, on the other hand, allows the optimization of cache allocation functions, and outperforms the state-of-the-art results for coded caching with non-uniform file popularities, as it will be shown in the sequel.*

C. Example

Here we highlight the differences between the delivery schemes outlined in Algorithms 2 and 3 through an example. For an arbitrary time slot $t \geq 2$, assume that $K^{(t)} = 5$ users are active: users 1, 2 and 3 (which joined the system in the current time slot) all request the first chunk of file W_2, while users 4 and 5 (which joined the system in the previous time slot) request the second chunk of W_1. Thus, the demand vector is $\mathbf{d}_t = \{21, 21, 21, 12, 12\}$, and $\mathcal{D}_t = \{W_{21}, W_{12}\}$.

The difference between Algorithms 2 and 3 lies in Part 1 and Part 2 corresponding to $z = 1$ and $z = 2$. For $z = 1$ in Algorithm 2, the following contents are delivered, each delivering the bits of the chunk requested by a user which have not been cached by any of the 5 users:

$$W^t_{21,\emptyset}, W^t_{21,\emptyset}, W^t_{21,\emptyset}, W^t_{12,\emptyset}, W^t_{12,\emptyset}. \tag{3}$$

On the other hand, Part 1 of Algorithm 3 delivers

$$W^t_{21,\emptyset}, W^t_{12,\emptyset}. \tag{4}$$

It is evident that Part 1 of Algorithm 3 delivers less number of bits than that of Algorithm 2 for $z = 1$.

With $z = 2$ in Algorithm 2, or equivalently Part 2 of Algorithm 3, the missing bits of each user's demand cached exclusively by each of the other 4 users is delivered. For $z = 2$ of Algorithm 2, the server delivers

$$W^t_{21,\{k\}} \overline{\oplus} W^t_{d_{k,t},\{1\}}, \quad \forall k \in [2:5], \tag{5a}$$
$$W^t_{21,\{k\}} \overline{\oplus} W^t_{d_{k,t},\{2\}}, \quad \forall k \in [3:5], \tag{5b}$$
$$W^t_{21,\{k\}} \overline{\oplus} W^t_{d_{k,t},\{3\}}, \quad \forall k \in [4:5], \tag{5c}$$
$$W^t_{12,\{5\}} \overline{\oplus} W^t_{12,\{4\}}. \tag{5d}$$

The coded contents delivered with (5a) enable user 1 to receive the bits of its demand cached exclusively by user k, while user k can also obtain the missing bits of its request cached exclusively by user 1, for $k \in [2:5]$. Also, coded contents delivered by (5b) enable user 2 to obtain the missing part of its requested chunk cached exclusively by user k, and so on so forth, for $k \in [3:5]$. Part 2.1 of Algorithm 3 delivers the same coded contents as in (5). On the other hand, with Part 2.2 of Algorithm 3 the following contents are delivered

$$W^t_{21,\{k\}} \overline{\oplus} W^t_{21,\{k+1\}}, \quad \forall k \in [4], \tag{6a}$$
$$W^t_{12,\{k\}} \overline{\oplus} W^t_{12,\{k+1\}}, \quad \forall k \in [4]. \tag{6b}$$

Note that among Part 2.1 and Part 2.2 of Algorithm 3 the one that delivers less number of bits in total is executed. The number of bits delivered in (6) is $R_{\mathrm{PCD}_2} = 4(T_{21,1}+T_{12,1})$. If $T_{12,1} \geq T_{21,1}$, the number of bits delivered in (5) is $R_{\mathrm{MAN}_2} = 3T_{21,1} + 7T_{12,1}$. It is trivial to see that $R_{\mathrm{PCD}_2} < R_{\mathrm{MAN}_2}$, i.e., Part 2 delivers less number of bits than $z = 2$ for Algorithm 2. On the other hand, if $T_{21,1} \geq T_{12,1}$, we have $R_{\mathrm{MAN}_2} = 9T_{21,1} + T_{12,1}$, and it follows that $R_{\mathrm{PCD}_2} < R_{\mathrm{MAN}_2}$.

Part 3 of Algorithm 3 performs the same as Algorithm 2 for $z \geq 3$; and hence, their performances are the same.

D. Average Delivery Rate

Here we present a closed-form expression for the achievable average delivery rate of the proposed coded caching scheme. For ease of presentation, we first introduce some notations:

- Let p^j denote the probability that a user watches the jth chunk. We have
$$p^j = \sum_{i=1}^{N} p_i p_{ij}, \quad \forall j \in [B]. \tag{7}$$
- For any time slot at the delivery phase, we have[2]
$$\Pr\{K_j = k\} = \sum_{a=k}^{A_{max}} P_A(a) \binom{a}{k} (p^j)^k (1-p^j)^{a-k}, \tag{8}$$
$\forall k \in \mathcal{A}, j \in [B]$, where we assume $0^0 = 1$.
- Let $\tilde{p}_{ij}$ denote the probability of a user requesting W_{ij} given that she is demanding the jth chunk of a file, i.e., $\tilde{p}_{ij} = p_i p_{ij}/p^j$, $\forall i \in [N], j \in [B]$. Note that we have $\sum_{i=1}^{N} \tilde{p}_{ij} = 1$. We refer to $\tilde{p}_{ij}$ as the *normalized popularity* of chunk W_{ij}, for $i \in [N]$ and $j \in [B]$.
- For a given set of $l \geq 1$ users, we define $g_{ij,(l,l')}$ as the number of bits of chunk W_{ij}, normalized by F/B, that have been cached by a subset of l' users among the l users, and not by any of the remaining $l - l'$ users, for $l' \in [l]$. Due to the law of large numbers, $g_{ij,(l,l')}$ is identical for any l users and any l' users among them, and we have
$$g_{ij,(l,l')} = (q_{ij})^{l'} (1-q_{ij})^{l-l'}, \quad \forall i \in [N], j \in [B], \tag{9}$$
with probability 1 as $F \to \infty$. Recall that q_{ij} is the caching probability for chunk W_{ij} as defined in Section III-A, which is identical across users.
- For a given time slot, let $\mathcal{S}_j$ be an l_j-element subset of users requesting the jth chunks of their demands, i.e., users in $\left[\sum_{h=1}^{j-1} K_h + 1 : \sum_{h=1}^{j} K_h\right]$, $j \in [B]$. We define $\mathcal{S}_{\text{sub}} \triangleq \bigcup_{j=1}^{B} \mathcal{S}_j$, and $\mathcal{S}_{\text{all}} \triangleq \bigcup_{j=1}^{B} \mathcal{S}_j = \left[\sum_{h=1}^{B} K_h\right]$. We denote by $\mathcal{D}_{\mathcal{S}_{\text{sub}}}$ the demand combination of the users in $\mathcal{S}_{\text{sub}}$, i.e., $\mathcal{D}_{\mathcal{S}_{\text{sub}}} \in \mathfrak{D}_{l_{[B]}}$, where $\mathfrak{D}_{l_{[B]}} \triangleq \{W_{11}, ..., W_{N1}\}^{l_1} \times \{W_{12}, ..., W_{N2}\}^{l_2} \times \cdots \times \{W_{1B}, ..., W_{NB}\}^{l_B}$. Let
$$\rho_{ij,(\mathcal{S}_{\text{all}},\mathcal{S}_{\text{sub}})} \triangleq \Pr\Bigg\{ \max_{W_{fh} \in \mathcal{D}_{\mathcal{S}_{\text{sub}}}} g_{fh,(\sum_{s=1}^{B} K_s, \sum_{s=1}^{B} l_s - 1)} = g_{ij,(\sum_{s=1}^{B} K_s, \sum_{s=1}^{B} l_s - 1)} \Bigg\}, \tag{10}$$
$\forall i \in [N], j \in [B]$, that is, $\rho_{ij,(\mathcal{S}_{\text{all}},\mathcal{S}_{\text{sub}})}$ is the probability that the maximum number of bits of a requested chunk by $\mathcal{S}_{\text{sub}}$ cached exclusively by $\sum_{s=1}^{B} l_s - 1$ users in $\mathcal{S}_{\text{sub}}$ (and not cached by the rest of the users in $\mathcal{S}_{\text{all}}$), which is identical for any $\sum_{s=1}^{B} l_s - 1$ users in $\mathcal{S}_{\text{sub}}$, is given by $g_{ij,(\sum_{s=1}^{B} K_s, \sum_{s=1}^{B} l_s - 1)}$. Since the file popularity and audience retention rates are identical among the users, the distribution of $\mathcal{D}_{\mathcal{S}_{\text{sub}}}$ only depends on $l_{[B]}$. Thus, for simplicity, we use $\mathcal{D}_{l_{[B]}}$ and $\rho_{ij,(K_{[B]},l_{[B]})}$ instead of $\mathcal{D}_{\mathcal{S}_{\text{sub}}}$ and $\rho_{ij,(\mathcal{S}_{\text{all}},\mathcal{S}_{\text{sub}})}$, respectively.

[2]Note that we remove the dependency of $K_j^{(t)}$ on t and replace it by K_j to make the notation valid for any time slot.

Theorem 1. *For the caching system described in Section II, and a given cache content distribution* $\mathbf{Q}$*, the following average delivery rate is achieved by the placement scheme presented in Section III-A followed by the* RAN *delivery scheme presented in Algorithm 1:*
$$R_{\text{RAN}}(P_A, \mathbf{p}, \mathbf{P}, \mathbf{Q}) = \sum_{j=1}^{B} \sum_{i=1}^{N} \sum_{k=0}^{A_{max}} \Pr\{K_j = k\} \cdot \left(1 - (1 - \tilde{p}_{ij})^K\right)(1 - q_{ij}). \tag{11}$$

Proof. The detailed proof can be found in Appendix A. □

Remark 4. *Given* $\mathbf{Q}$*, consider the uncoded caching and delivery scheme, shortly referred to as Uncoded, where each user caches the same* $q_{ij}F/B$ *bits from chunk* W_{ij}*,* $i \in [N]$*,* $j \in [B]$*, in the placement phase. At each time slot of the delivery phase, the server sends the missing* $(1-q_{ij})F/B$ *bits of chunk* W_{ij} *if it is requested. We note that for any demand combination, the Uncoded scheme sends the same number of bits as the* RAN *delivery scheme, for the placement scheme described in Section III-A, which results in the same average delivery rate given in* (11)*.*

Theorem 2. *For the caching system described in Section II, and a given cache content distribution* $\mathbf{Q}$*, the following average delivery rate is achieved by the placement scheme presented in Section III-A followed by the* MAN *delivery scheme presented in Algorithm 2:*
$$R_{\text{MAN}}(P_A, \mathbf{p}, \mathbf{P}, \mathbf{Q}) = \sum_{k_{[B]} \in \mathcal{A}^B} \prod_{s=1}^{B} \Pr\{K_s = k_s\} \cdot \left(\sum_{l_{[B]} \in [0:k_1] \times \cdots \times [0:k_B]} \prod_{j=1}^{B} \binom{k_j}{l_j} \cdot \sum_{j=1}^{B} \sum_{i=1}^{N} \rho'_{ij,(k_{[B]},\ l_{[B]})} g_{ij,(\sum_{s=1}^{B} k_s, \sum_{s=1}^{B} l_s - 1)} \right), \tag{12}$$
where
$$\rho'_{ij,(k_{[B]},\ l_{[B]})} \triangleq \frac{\rho_{ij,(k_{[B]},\ l_{[B]})}}{\sum_{f=1}^{N} \sum_{h=1}^{B} \mathbb{1}\left\{ g_{fh,(\sum_{s=1}^{B} k_s, \sum_{s=1}^{B} l_s - 1)} = g_{ij,(\sum_{s=1}^{B} k_s, \sum_{s=1}^{B} l_s - 1)} \right\}}. \tag{13}$$

Proof. The detailed proof can be found in Appendix B. □

Recall that $g_{ij,(\sum_{s=1}^{B} k_s, \sum_{s=1}^{B} l_s - 1)}$ represents the approximate number of bits of chunk W_{ij} that are cached exclusively by $\sum_{s=1}^{B} l_s - 1$ users among $\sum_{s=1}^{B} k_s$ users. Also, $\rho_{ij,(k_{[B]},\ l_{[B]})}$ denotes the probability that $g_{ij,(\sum_{s=1}^{B} k_s, \sum_{s=1}^{B} l_s - 1)}$, i.e., the expected number of bits of W_{ij} cached by $\sum_{s=1}^{B} l_s - 1$ users, is the number of bits sent to a subset of $z = \sum_{s=1}^{B} l_s$ users by Algorithm 2, among which l_j users request the jth chunks of their demands. The above upper bound is derived by summing over the expected number of bits sent to any subset of users by Algorithm 2, averaged over all demand combinations.

Remark 5. *We remark that when $B = 1$ and $\mathcal{A} = [K]$, the considered caching problem reduces to the one with non-uniform file popularities [11]. Rate $\min\{R_{\rm RAN}(P_A,\mathbf{p},\mathbf{P},\mathbf{Q}), R_{\rm MAN}(P_A,\mathbf{p},\mathbf{P},\mathbf{Q})\}$ can be achieved by performing the scheme resulting in a smaller delivery rate among the* RAN *and* MAN *schemes for given* $\mathbf{Q}$*. For a given* $\mathbf{Q}$*, where $q_{ij} = q_{fh}$ for some $ij \neq fh, i,f \in [N], j,h \in [B]$, (such that $g_{fh,(\sum_{s=1}^B k_s, \sum_{s=1}^B l_s)} = g_{ij,(\sum_{s=1}^B k_s, \sum_{s=1}^B l_s)}$ for any $k_{[B]}$ and $l_{[B]}$), it provides a tighter upper bound than the one in [11, Theorem 1] due to the denominator in* (13)*. However, the optimization of cache allocation over the average delivery rate will ensure that the cache capacities allocated to different files are distinct. Hence, with optimal cache allocation the upper bound in [11, Theorem 1] can be arbitrarily close to $\min\{R_{\rm RAN}(P_A,\mathbf{p},\mathbf{P},\mathbf{Q}), R_{\rm MAN}(P_A,\mathbf{p},\mathbf{P},\mathbf{Q})\}$.*

Theorem 3. *For the caching system described in Section II, and a given cache content distribution* $\mathbf{Q}$*, the following average delivery rate is achieved by the placement scheme presented in Section III-A followed by the PCD delivery scheme outlined in Algorithm 3:*

$$R_{\rm PCD}(P_A,\mathbf{p},\mathbf{P},\mathbf{Q}) \triangleq R_{\rm MAN}(P_A,\mathbf{p},\mathbf{P},\mathbf{Q}) - \Delta\varphi_1(P_A,\mathbf{p},\mathbf{P},\mathbf{Q}) - \Delta\varphi_2(P_A,\mathbf{p},\mathbf{P},\mathbf{Q}), \tag{14a}$$

where

$$\Delta\varphi_1(P_A,\mathbf{p},\mathbf{P},\mathbf{Q}) \triangleq \sum_{k_{[B]}\in\mathcal{A}^B}\prod_{s=1}^B \Pr\{K_s=k_s\}\left(\sum_{j=1}^B k_j \sum_{i=1}^N \tilde{p}_{ij} g_{ij,(\sum_{s=1}^B k_s,0)} - \sum_{j=1}^B\sum_{i=1}^N \left(1-(1-\tilde{p}_{ij})^{k_j}\right) g_{ij,(\sum_{s=1}^B k_s,0)}\right), \tag{14b}$$

and

$$\Delta\varphi_2(P_A,\mathbf{p},\mathbf{P},\mathbf{Q}) \triangleq \sum_{k_{[B]}\in\mathcal{A}^B}\prod_{s=1}^B \Pr\{K_s=k_s\} \max\Bigg\{ \sum_{j=1}^B \binom{k_j}{2} \sum_{i=1}^N \rho'_{ij,(k_{[B]},\ (\mathbf{0},l_j=2,\mathbf{0}))} g_{ij,(\sum_{s=1}^B k_s,1)} + \sum_{j_1=1}^B k_{j_1} \sum_{j_2=j_1+1}^B k_{j_2} \sum_{i=1}^N \sum_{j=1}^B \rho'_{ij,(k_{[B]},(\mathbf{0},l_{j_1}=1,\mathbf{0},l_{j_2}=1,\mathbf{0}))} g_{ij,(\sum_{s=1}^B k_s,1)} - \sum_{j=1}^B\sum_{i=1}^N (\sum_{s=1}^B k_s - 1)\left(1-(1-\tilde{p}_{ij})^{k_j}\right) g_{ij,(\sum_{s=1}^B k_s,1)}, 0\Bigg\}, \tag{14c}$$

where $(\mathbf{0}, l_j = 2, \mathbf{0})$ is a B-element vector, whose jth element is 2 while the rest are zero, for some $j \in [B]$, while $(\mathbf{0}, l_{j_1} = 1, \mathbf{0}, l_{j_2} = 1, \mathbf{0})$ is a B-element vector, whose j_1-th and j_2-th elements are 1 while the rest are zero, for some $j_1, j_2 \in [B]$, $j_1 \neq j_2$.

Proof. Observe that the missing bits sent in PART 1 and PART 2 of Algorithm 3 are sent by the delivery scheme in Algorithm 2 for $z = 1$ and $z = 2$, respectively; and PART 3 is the same as the delivery scheme in Algorithm 2 for $z > 2$. We point out here that $\Delta\varphi_1(P_A,\mathbf{p},\mathbf{P},\mathbf{Q})$ is the difference between the average number of bits sent by PART 1 of Algorithm 3 and those sent by the delivery scheme in Algorithm 2 for $z = 1$, while $\Delta\varphi_2(P_A,\mathbf{p},\mathbf{P},\mathbf{Q})$ is the difference between the average number of bits sent by PART 2 of Algorithm 3 and those sent by the delivery scheme in Algorithm 2 for $z = 2$. Hence, we have the delivery rate achieved by the delivery scheme in Algorithm 3 as in (14a). The detailed proofs of (14b) and (14c) are provided in Appendix C. □

The value of $\rho_{ij,(k_{[B]},l_{[B]})}$ can be calculated as follows. We define, $\forall D \in \mathfrak{D}_{l_{[B]}}$,

$$Y_{k_{[B]},l_{[B]}}(D) \triangleq \max_{W_{fh}\in D} g_{fh,(\sum_{s=1}^B k_s, \sum_{s=1}^B l_s - 1)}. \tag{15}$$

Let

$$\mathfrak{D}'_{l_{[B]},ij} \triangleq \left\{ D \in \mathfrak{D}_{l_{[B]}} : Y_{k_{[B]},l_{[B]}}(D) \leq g_{ij,(\sum_{s=1}^B k_s, \sum_{s=1}^B l_s - 1)} \right\},$$

i.e., $\mathfrak{D}'_{l_{[B]},ij}$ is the set of all elements D in $\mathfrak{D}_{l_{[B]}}$ such that $Y_{k_{[B]},l_{[B]}}(D) \leq g_{ij,(\sum_{s=1}^B k_s, \sum_{s=1}^B l_s - 1)}$. Similarly, let

$$\mathfrak{D}''_{l_{[B]},ij} \triangleq \left\{ D \in \mathfrak{D}_{l_{[B]}} : Y_{k_{[B]},l_{[B]}}(D) < g_{ij,(\sum_{s=1}^B k_s, \sum_{s=1}^B l_s - 1)} \right\},$$

and

$$\mathfrak{D}'''_{l_{[B]},ij} \triangleq \left\{ D \in \mathfrak{D}_{l_{[B]}} : Y_{k_{[B]},l_{[B]}}(D) = g_{ij,(\sum_{s=1}^B k_s, \sum_{s=1}^B l_s - 1)} \right\}.$$

We have $\mathfrak{D}'''_{l_{[B]},ij} = \mathfrak{D}'_{l_{[B]},ij} \setminus \mathfrak{D}''_{l_{[B]},ij}$. It follows that

$$\sum\nolimits_{D\in\mathfrak{D}'_{l_{[B]},ij}} \Pr\left\{\mathcal{D}_{l_{[B]}} = D\right\} = \prod_{h=1}^B \left(\sum_{W_f\in\mathcal{F}: g_{fh,(\sum_{s=1}^B k_s, \sum_{s=1}^B l_s - 1)} \leq g_{ij,(\sum_{s=1}^B k_s, \sum_{s=1}^B l_s - 1)}} \tilde{p}_{fh} \right)^{l_h},$$

that is, the probability that a demand combination $\mathcal{D}_{l_{[B]}}$ falls in the set $\mathfrak{D}'_{l_{[B]},ij}$, i.e., $Y_{k_{[B]},l_{[B]}}(\mathcal{D}_{l_{[B]}}) \leq g_{ij,(\sum_{s=1}^B k_s, \sum_{s=1}^B l_s - 1)}$, is the probability that each requested chunk, $W_{fh} \in \mathcal{D}_{l_{[B]}}$, is associated with $g_{fh,(\sum_{s=1}^B k_s, \sum_{s=1}^B l_s - 1)}$ no greater than $g_{ij,(\sum_{s=1}^B k_s, \sum_{s=1}^B l_s - 1)}$, given that there are l_h requests of the h-th chunks, $\forall h \in [B]$. Similarly,

$$\sum\nolimits_{D\in\mathfrak{D}'_{l_{[B]},ij}} \Pr\left\{\mathcal{D}_{l_{[B]}} = D\right\} = \prod_{h=1}^B \left(\sum_{W_f\in\mathcal{F}: g_{fh,(\sum_{s=1}^B k_s, \sum_{s=1}^B l_s - 1)} < g_{ij,(\sum_{s=1}^B k_s, \sum_{s=1}^B l_s - 1)}} \tilde{p}_{fh} \right)^{l_h},$$

i.e., the probability that the value of $g_{fh,(\sum_{s=1}^B k_s, \sum_{s=1}^B l_s - 1)}$ for each requested chunk, $W_{fh} \in \mathcal{D}_{l_{[B]}}$, is less than $g_{ij,(\sum_{s=1}^B k_s, \sum_{s=1}^B l_s - 1)}$, given that there are l_h requests of the h-th chunks, $\forall h \in [B]$ (note that it is "no larger than" in the

case of (16)). According to the definition of $\rho_{ij,(k_{[B]},l_{[B]})}$ given in (10), we derive

$$\begin{aligned}\rho_{ij,(k_{[B]},\ l_{[B]})} &= \sum\nolimits_{D\in\mathfrak{D}'''_{l_{[B]},ij}} \Pr\left\{\mathcal{D}_{l_{[B]}}=D\right\}\\ &= \sum\nolimits_{D\in\mathfrak{D}'_{l_{[B]},ij}} \Pr\left\{\mathcal{D}_{l_{[B]}}=D\right\}\\ &\quad - \sum\nolimits_{D\in\mathfrak{D}''_{l_{[B]},ij}} \Pr\left\{\mathcal{D}_{l_{[B]}}=D\right\}. \quad (16)\end{aligned}$$

Thus, $\rho_{ij,(k_{[B]},\ l_{[B]})}$ can be easily calculated by sorting $\left\{g_{fh,(\sum_{s=1}^{B}k_s,\sum_{s=1}^{B}l_s-1)}, f\in[N], h\in[B]\right\}$.

Remark 6. *It is trivial to see that* $\Delta\varphi_1(P_A,\mathbf{p},\mathbf{P},\mathbf{Q})\geq 0$*, and the equality holds only when* $\mathcal{A}=\{1\}$*, and* $\Delta\varphi_2(P_A,\mathbf{p},\mathbf{P},\mathbf{Q})\geq 0$ *according to* (14c)*. Hence, we can conclude that the PCD scheme achieves a lower average delivery rate than the* MAN *scheme when* $\mathcal{A}\neq\{1\}$*, which will further be validated by the numerical results.*

E. Cache Allocation

Here we derive the best cache content distribution $\mathbf{Q}$ by solving the following optimization problem:

$$\min R(\mathbf{Q}) \quad (17a)$$

$$\text{s.t.}\sum\nolimits_{i,j} q_{ij}=MB, \quad (17b)$$

where the objective is to minimize the average delivery rate over all possible demand combinations while the cache capacity constraint at each user is satisfied with equality. We consider $R_{\text{MAN}}(\mathbf{Q})$ and $R_{\text{PCD}}(\mathbf{Q})$ as the objective functions, for the MAN and PCD schemes, respectively. We note that this optimization problem is non-linear and non-convex; and hence, is highly intractable to be solved exactly. We use sequential quadratic programming (SQP) to solve it numerically. The corresponding solution will be referred to as the HeCA.

In practice, however, there will be a large number of files in the library, and each video file can be partitioned into many chunks. In that case, optimizing $\mathbf{Q}$ over all the chunks in the library requires high computational complexity. To reduce complexity, the authors in [11] employ a simplified cache placement method, where users cache the same fraction of (randomly selected) packets from the most popular files. Here we present an alternative low-complexity solution to HeCA, referred to as HoCA, in which the same number of (randomly selected) bits from the most popular chunks are cached by the users; that is, we have

$$q_{ij}=\begin{cases}q, & \text{if } p_ip_{ij}\geq\overline{n},\\ 0, & \text{otherwise,}\end{cases} \quad (18)$$

where $q\in(0,1]$ and $\overline{n}$ are the two parameters to be chosen to satisfy $\sum_{i=1}^{N}\sum_{j=1}^{B}q_{ij}=MB$. We denote the cache content distribution given by (18) as a function of q, i.e., $\mathbf{Q}(q)$. The optimization of q can be expressed as $q^*\triangleq\operatorname{argmin}R(\mathbf{Q}(q))$, which can be computed through one-dimensional search. The comparison of the presented caching schemes through numerical simulations is relegated to Section V.

IV. Lower Bound

Here, we present a lower bound on the average delivery rate-cache capacity tradeoff $R(M)$, derived by assuming that some of the requested chunks are served by a genie at no transmission cost, inspired by the lower bound in [11]. Accordingly, the problem is relaxed to a caching problem with uniform file popularity, whose delivery rate can be bounded through cut-set arguments.

Theorem 4. *For the caching problem described in Section II,* $R(M)$ *is lower bounded by*

$$\begin{aligned}R(M)\geq R^*(M)\triangleq \sum_{k_{[B]}\in\mathcal{A}^B}\prod_{j=1}^{B}\Pr\{K_j=k_j\}\cdot\\ \max_{n_{[B]},v_{[B]},\widetilde{z}_{[B]}}\left\{\left(\prod_{j=1}^{B}f'_j(k_j,n_j,v_j)\right)\left(\prod_{j=1}^{B}f''_j(n_j,v_j,\widetilde{z}_j)\right)\cdot\right.\\ \left.\max_{\substack{z_j\in[\lceil\min\{\widetilde{z}_j,v_j\}\rceil]\\ j\in[B]}}\left\{\sum_{j=1}^{B}z_j\left(1-\frac{MB}{\min\limits_{j\in[B]}\lfloor\frac{n_j}{z_j}\rfloor}\right)\right\}\right\}, \quad (19)\end{aligned}$$

where $n_j\in[N]$*,* $v_j\in(0,k_jn_jr_{n_jj}]$*,* $\widetilde{z}_j\in(0,f(n_j,v_j)]$*,* $j\in[B]$*, and*

$$f(n_j,v_j)\triangleq n_j\left(1-\left(1-\frac{1}{n_j}\right)^{v_j}\right), \quad (20)$$

$$f'_j(k_j,n_j,v_j)\triangleq 1-\exp\left(-\frac{(k_jn_jr_{n_jj}-v_j)^2}{2k_jn_jr_{n_jj}}\right), \quad (21)$$

$$f''_j(n_j,v_j,\widetilde{z}_j)\triangleq 1-\exp\left(-\frac{(f(n_j,v_j)-\widetilde{z}_j)^2}{2f(n_j,v_j)}\right), \quad (22)$$

and $r_{1j},...,r_{Nj}$ *is an ordered permutation of* $\{\tilde{p}_{1j},...,\tilde{p}_{Nj}\}$*, such that* $r_{1j}\geq\cdots\geq r_{Nj}$*,* $\forall j\in[B]$*.*

Proof. The detailed proof can be found in Appendix D. □

Here, n_j is a parameter that controls which chunk is served by the genie. In particular, considering that user k demands the jth chunk of its request, if the requested chunk has a normalized popularity lower than r_{n_jj}, i.e., $\tilde{p}_{d_{k,t}}<r_{n_jj}$, it is served by a genie at no transmission cost; otherwise, i.e., if $\tilde{p}_{d_{k,t}}\geq r_{n_jj}$, it is served by a genie with probability $1-r_{n_jj}/\tilde{p}_{d_{k,t}}$; that is, the server has to transmit the required jth chunk to this user over the shared link with probability $r_{n_jj}/\tilde{p}_{d_{k,t}}$. Given k_j users requesting the jth chunks and parameter n_j, $f'_j(k_j,n_j,v_j)$ represents the probability that v_j users among k_j require service from the server, and the rest k_j-v_j users are served by the genie. $f''_j(n_j,v_j,\widetilde{z}_j)$ denotes the probability that $\widetilde{z}_j$ distinct jth chunks are requested by these v_j users, which require service from the server given parameter n_j. Following the cut-set approach, the average delivery rate to deliver z_j distinct demands from a library of n_j chunks with uniform popularity, $j\in[B]$, is lower bounded by $\sum_{j=1}^{B}z_j\left(1-MB/\min_{j\in[B]}\lfloor n_j/z_j\rfloor\right)$. As we will see next the above lower bound is loose, due both to the looseness of the cut-set bound, and the genie aided assumption.

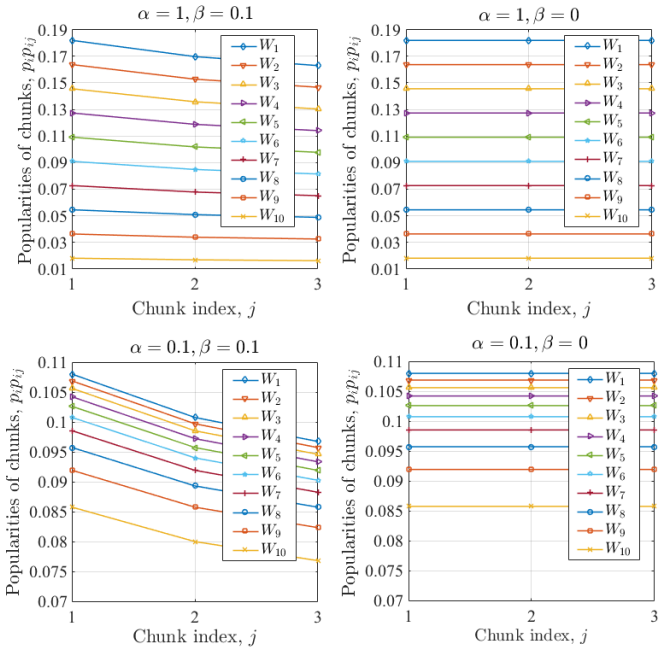

Fig. 2. The popularity of video chunks W_{ij}, i.e., $p_i p_{ij}$ given different values of α and β.

V. Numerical Results

In this section, we numerically evaluate the average delivery rate achieved by the two coded delivery schemes, i.e., MAN and PCD, with both cache allocation strategies, HoCA and HeCA, and compare with RAN, the uncoded caching as well as the lower bound. In uncoded caching, each user fully caches as many of the most popular chunks as possible to fill its cache capacity. We consider $N = 10$ video files in the library, each consists of $B = 3$ chunks of equal size. We assume that the popularity of files follows a Zipf power law with parameter α [21], in which case we have $p_i = (6-i)^\alpha / \left(\sum_{f=1}^{N} f^\alpha\right)$, for $i \in [5]$, and the audience retention rates of the video files follow a Zipf-like distribution as well [22], i.e., $p_{ij} = j^{-\beta_i}$, for $i \in [5], j \in [3]$, with parameter $\beta_i \geq 0$. The larger β_i implies a shorter average watching time for file W_i. We set β_i to be identical for all the files, i.e., $\beta_i = \beta$, $\forall i \in [N]$, and the corresponding popularity of chunks, i.e., $p_i p_{ij}$, $i \in [5]$, $j \in [3]$, are presented in Fig. 2 for different values of α and β. It shows that a higher α results in a larger difference in file popularities. Moreover, with a larger β, the retention rates of the video files decrease more quickly with the chunk index.

It is not tractable to get an analytical form of $\mathbf{Q}$ derived by minimization of $R_{\text{MAN}}(\mathbf{Q})$ or $R_{\text{PCD}}(\mathbf{Q})$ which is a non-convex problem without an appealing structure. In Fig. 3, we show the caching distribution $\mathbf{Q}$ derived by numerically minimizing $R_{\text{PCD}}(\mathbf{Q})$ under the above setting assuming there are 5 or 15 demands arriving at each time slot, i.e. $\mathcal{A} = \{5\}$ or $\{15\}$ while $M = 1$. It can be observed that although the popularity of each chunk is the same, the optimized caching distribution varies with the number of demands arriving each time slot. When $\mathcal{A} = \{15\}$, the minimization of $R_{\text{PCD}}(\mathbf{Q})$ imposes equal allocation of cache capacity among chunks, except the three least popular ones. This is due to the fact that when the number of new damands per time slot is relatively large, the probability that a chunk is requested by at least one user is high even when the popularity of this chunk is relatively low. Hence, to benefit from coded delivery, it is preferable to maintain the symmetry in cache allocation. On the contrary, when $\mathcal{A} = \{5\}$, we cache only the most popular chunks fully. When the number of new demands is relatively small, the demands are more likely to concentrate on the most popular chunks. Therefore, caching those chunks will maximize the caching gain.

Here, we consider $\mathcal{A} = \{15\}$; that is, exactly 15 new demands arrive at each time slot. It is verified in Fig. 4 that, the RAN and Uncoded schemes have the same performance. We also observe from Fig. 4 that both PCD and MAN significantly reduce the average delivery rate compared to the Uncoded and RAN schemes, and the improvement increases with the cache capacity. We can see that the PCD scheme notably outperforms MAN when the cache capacity is small, as PCD is more efficient in delivering the bits that either have not been cached by any user, or have been cached exclusively by one user. An interesting observation is that, both PCD and MAN achieve almost the same performance with either of the two cache allocation schemes, HoCA and HeCA. This implies that caching as many of the most popular chunks as possible can be sufficient to fully exploit the cache capacities. However, slight improvement of HoCA over HeCA can be observed in the zoomed-in subfigures in Fig. 4. We can also observe that a larger α results in a smaller average delivery rate since the users tend to request the most popular files, and caching these files is more efficient. In contrast, a smaller β increases the average delivery rate since users tend to continue watching, which increases the overall demand. We also note that the gap between the lower bound and the achievable delivery rate remains significant, which calls for a tighter lower bound.

We evaluate the effect of the asynchronous arrival of demands by considering two scenarios: in the first scenario, 15 new users arrive at each time slot as in Fig. 4; while in the second one, 45 new users arrive at every three time slots, while the demands are asynchronous in the first scenario, they are synchronized in the second as all the active users watch the same chunk. The average delivery rates achieved by PCD-HeCA are shown in Fig. 5 for the two scenarios, labeled as *PCD-async* and *PCD-sync*. We see that PCD-sync has remarkably lower average delivery rates than PCD-async when the cache capacity is small, since there are less distinct demands in each time slot. However, as the cache capacity increases, the effect of distinct demands is compensated since coded delivery can create multicasting opportunities by exploiting the cached contents. Hence, we can conclude from Fig. 5 that larger cache capacities are needed to observe the benefits of coded delivery in the more realistic setting of asynchronous user demands.

In Fig. 6, we compare the performance of PCD with the RAP-GCC scheme in [11] and the scheme proposed in [23], referred to as SIM scheme, which, to the best of our knowledge, are the only results in the literature on the average delivery rate considering heterogeneous file popularities. We set $B = 1$, such that the partial caching problem studied in this paper reduces to the one in [11]. The RAP-GCC scheme

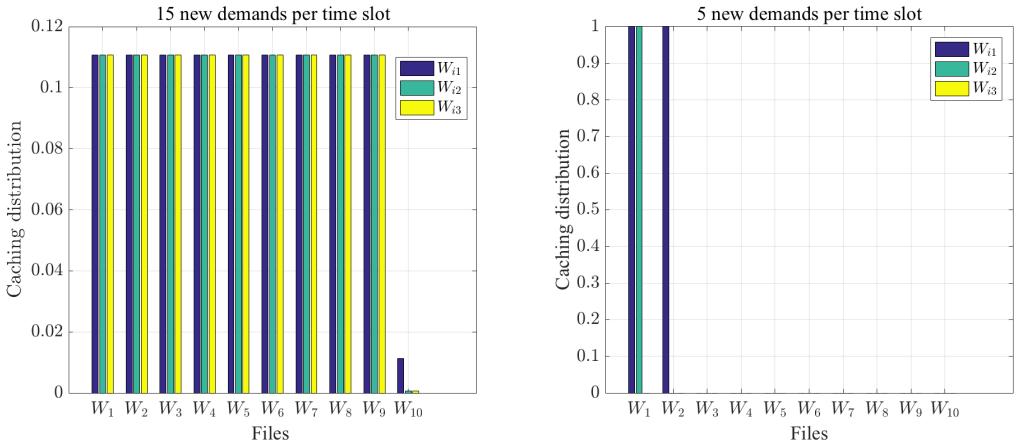

Fig. 3. The caching distribution derived by minimizing $R_{\mathrm{PCD}}(\mathbf{Q})$ given, $M = 1$.

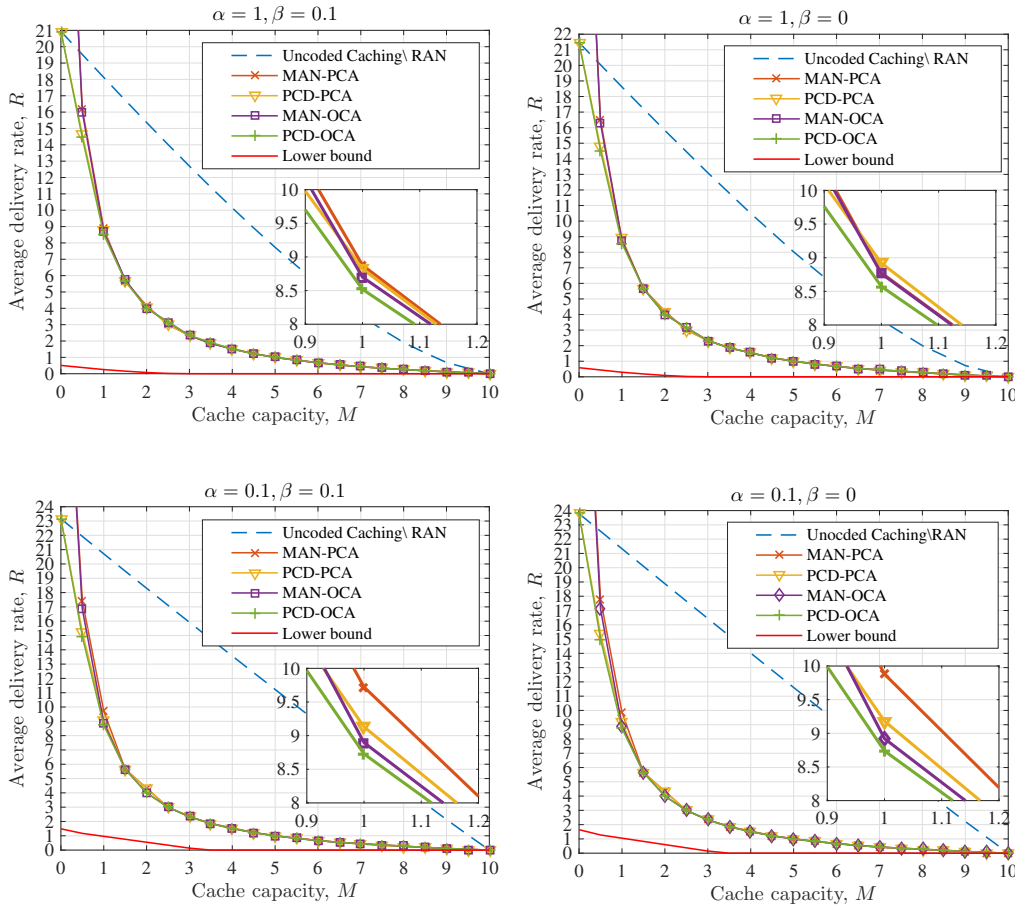

Fig. 4. Comparison between PCD, MAN, uncoded caching and the lower bound given different values of α and β.

consists of cache placement, in which each user caches a fraction of each file selected at random, and linear coded delivery based on chromatic number index coding. For the fairness of comparison, we optimize the cache content distribution for the RAP-GCC scheme as well. The SIM scheme simply allocates all the cache capacity to the N_1 most popular files equally when N_1 is optimized to minimize the average delivery rate, which is similar to the HoCA scheme in this paper and RFLU scheme in [11]. We can observe in Fig. 6 that both PCD-HeCA and RAP-GCC schemes remarkably improve over the SIM scheme. It is also notable that PCD outperforms RAP-GCC, and as α becomes larger, i.e, the popularity distribution of the files becomes more skewed, the gap between the two schemes increases slightly. We note here that PCD outperforms RAP for a larger range of M values compared to Fig. 4. The benefits of PCD come from sending the bits that are cached

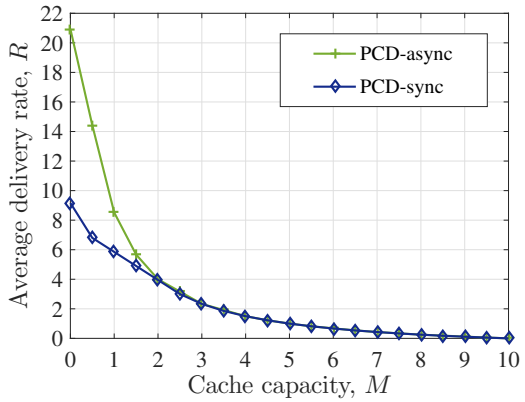

Fig. 5. Comparison between the asynchronous and synchronous demand arrival scenarios, $\alpha = 1$ and $\beta = 0.1$.

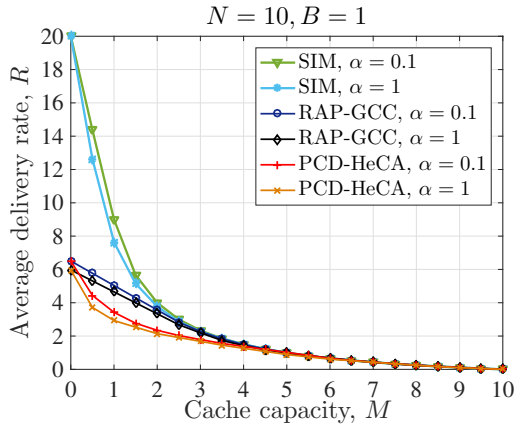

Fig. 6. Comparison between PCD with HeCA and RAP-GCC with $\alpha = 0.1$ and $\alpha = 1$.

by none or only one of the users more efficiently compared to RAP. Note that there are 15 active users in the setting of Fig. 6 while there are nearly 45 (a few of them may be inactive because of an early quit) in the setting of Fig. 4. With a larger number of users in the system, the number of bits that are cached by none or only one of the users is more likely to be smaller. Hence, the benefit of PCD compared to MAN is less significant in Fig. 4.

VI. Conclusions

We have studied content caching and coded delivery in a more realistic system model, allowing asynchronous demand arrivals, and taking into account the audience retention rates of the video files; that is, we allow the users to dynamically join the system, place a request, consume a random portion of the request, and leave the system at a random time. We assume that each video file in the library consists of a number of chunks of the same size, and the audience retention rate is modeled as the heterogeneous popularity of the chunks of each file. We proposed a coded caching scheme that allocates users' cache capacities to different chunks, depending on their popularities. We then evaluated the average delivery rate over all possible demand combinations. We employed two different methods for cache allocation, namely, the numerically optimized cache allocation scheme HeCA, and a low complexity popularity-based cache allocation scheme HoCA. The numerical results showed a significant improvement with the proposed scheme over uncoded caching in terms of the average delivery rate, or the extension of other known delivery methods to the asynchronous scenario. We have also derived an information theoretic lower bound on the average delivery rate.

Appendix A
Proof of Theorem 1

Since each user requesting chunk W_{ij} already has $q_{ij}F/B$ bits of it cached, according to [8, Appendix A], at most $(1-q_{ij})F/B + o(F/B)$ bits are necessary to enable all the users requesting W_{ij} to decode it, for $i \in [N]$ and $j \in [B]$. The probability that chunk W_{ij} is requested by at least one user at time slot t is given by:

$$\Pr\{W_{ij} \in \mathcal{D}_t\} = \sum\nolimits_{k\in\mathcal{A}} \Pr\{K_j = k\}\Big(1-(1-\tilde{p}_{ij})^k\Big). \tag{23}$$

By summing over $i \in [N]$ and $j \in [B]$, and ignoring the $o(F/B)$ term, we complete the proof:

$$R_{\text{RAN}}(P_A, \mathbf{p}, \mathbf{P}, \mathbf{Q}) = \sum_{j=1}^{B}\sum_{i=1}^{N}\sum_{k\in\mathcal{A}} \Pr\{K_j = k\}\Big(1-(1-\tilde{p}_{ij})^K\Big)(1-q_{ij}). \tag{24}$$

Appendix B
Proof of Theorem 2

Recall that K_j is the number of users demanding the jth chunks of their requested files at time slot t, and these K_j users are indexed with $\left[K'_{j-1}+1 : K'_j\right]$, where $K'_j \triangleq \sum_{s=1}^{j} K_s$ and $K'_0 \triangleq 0$, for $j \in [B]$. Similar to the proof in [11, Appendix A], the average number of bits (normalized by F/B) sent by the MAN scheme over all possible demand combinations is given by

$$R^t_{\text{MAN}}(K_{[B]}) = \mathbb{E}\left[\sum\nolimits_{z=0}^{K^{(t)}} \sum\nolimits_{\mathcal{P}\subset[K^{(t)}],|\mathcal{P}|=z} \max_{k\in\mathcal{P}} \left|W^t_{d_{k,t},\mathcal{P}\setminus\{k\}}\right| \middle| C\right] \tag{25a}$$

$$= \sum\nolimits_{\mathbf{d}_t\in[N]^{K_1}\times[N]^{K_2}\times\cdots\times[N]^{K_B}} \prod_{j=1}^{B}\prod_{k=1}^{K_j} \tilde{p}_{d_{K'_{j-1}+k,t}} \left(\sum_{l_{[B]}\in[0:K_1]\times\cdots\times[0:K_B]} \sum_{\substack{\mathcal{P}_1\subset[K'_0+1:K'_1]:\\ |\mathcal{P}_1|=l_1}} \sum_{\substack{\mathcal{P}_2\subset[K'_1+1:K'_2]:\\ |\mathcal{P}_2|=l_2}} \cdots \sum_{\substack{\mathcal{P}_B\subset[K'_{B-1}+1:K'_B]:\\ |\mathcal{P}_B|=l_B}} \max_{\substack{k\in\mathcal{P}:\\ \mathcal{P}=\bigcup_{s=1}^{B}\mathcal{P}_s}} \left|W^t_{d_{k,t},\mathcal{P}\setminus\{k\}}\right| \right) \tag{25b}$$

$$= \sum\nolimits_{l_{[B]}\in[0:K_1]\times\cdots\times[0:K_B]} \prod\nolimits_{j=1}^{B} \binom{K_j}{l_j} \sum\nolimits_{(d_{1,t},\ldots,d_{l_1,t})\in[N]^{l_1}} \cdots \sum\nolimits_{(d_{K'_{B-1}+1,t},\ldots,d_{K'_{B-1}+l_B,t})\in[N]^{l_B}}$$

$$\prod_{j=1}^{B}\prod_{k=1}^{l_j} \tilde{p}_{d_{K'_{j-1}+k,t}} \sum_{\substack{k\in\mathcal{P}:\\ \mathcal{P}=\bigcup_{s=1}^{B}\mathcal{P}_s,\mathcal{P}_s=[K'_{s-1}+1:K'_{s-1}+l_s]}} \frac{\mathbb{1}\left\{k=\underset{h\in\mathcal{P}}{\operatorname{argmax}}\left|W^t_{d_{h,t},\mathcal{P}\backslash\{h\}}\right|\right\}}{\sum_{h\in\mathcal{P}}\mathbb{1}\left\{\left|W^t_{d_{h,t},\mathcal{P}\backslash\{h\}}\right|=\left|W^t_{d_{k,t},\mathcal{P}\backslash\{k\}}\right|\right\}}\cdot\left|W^t_{d_{k,t},\mathcal{P}\backslash\{k\}}\right| \tag{25c}$$

$$= \sum\nolimits_{l_{[B]}\in[0:K_1]\times\cdots\times[0:K_B]} \prod\nolimits_{j=1}^{B} \binom{K_j}{l_j} \sum\nolimits_{i=1}^{N}\sum\nolimits_{j=1}^{B}$$

$$\frac{\Pr\left(\max\limits_{W_{fh}\in\bigcup_{s=1}^{B}\{W_{d_{K'_{s-1}+1,t}},\ldots,W_{d_{K'_{s-1}+l_s,t}}\}}\left|W^t_{fh,[\sum_{s=1}^{B}l_s-1]}\right|=\left|W^t_{ij,[\sum_{s=1}^{B}l_s-1]}\right|\right)}{\sum_{f=1}^{N}\sum_{h=1}^{B}\mathbb{1}\left\{\left|W^t_{fh,[\sum_{s=1}^{B}l_s-1]}\right|=\left|W^t_{ij,[\sum_{s=1}^{B}l_s-1]}\right|\right\}}\left|W^t_{ij,[\sum_{s=1}^{B}l_s-1]}\right| \tag{25d}$$

where C is the realization of cache contents for a fixed cache content distribution $\mathbf{Q}$; (25b) is derived by finding the expectation over all possible demand realizations $\mathbf{d}_t$ given $K_1,...,K_B$. Note that the probability of any demand combination $\mathbf{d}_t \in [N]^{K_1}\times[N]^{K_2}\times\cdots\times[N]^{K_B}$ is thus $\prod_{j=1}^{B}\prod_{k=1}^{K_j}\tilde{p}_{d_{K'_{j-1}+k,t}}$. (25b) also specifies the number of users in $\mathcal{P}$ requesting different chunks, i.e., $l_1,...,l_B$, where l_j is the number of users in $\mathcal{P}$ demanding the jth chunks, for $j\in[B]$. Notice that $\max_{k\in\mathcal{P}:\mathcal{P}=\bigcup_{s=1}^{B}\mathcal{P}_s}\left|W^t_{d_{k,t},\mathcal{P}\backslash\{k\}}\right|$ depends only on the demands of users in $\mathcal{P}$. (25c) follows by first changing the order of the summation, which is to choose a set of $\sum_{s=1}^{B}l_s$ users first, among which l_j users request their jth chunks, for $j\in[B]$, and then take the expectation of the number of bits sent to this set of users over all possible demand combinations. Note that, due to the symmetry across users, for a given l_j, $j\in[B]$, any l_j users among K_j can be considered. Henceforth, for any $(l_1,...,l_B)\in[0:K_1]\times\cdots\times[0:K_B]$, (25c) only considers the first l_j users, i.e., users $K'_{j-1}+1,...,K'_{j-1}+l_j$, among the K_j users demanding the jth chunks, for $j\in[B]$, without any loss of accuracy. Writing the expectation with regards to each chunk yields (25d), where we note that if $\sum_{s=1}^{B}l_s=0$, $\left|W^t_{ij,[\sum_{s=1}^{B}l_s-1]}\right|=0$, $\forall i\in[N],\forall j\in[B]$. We emphasize that

$$\Pr\Bigg(\max\nolimits_{W_{fh}\in\bigcup_{s=1}^{B}\{W_{d_{K'_{s-1}+1,t}},\ldots,W_{d_{K'_{s-1}+l_s,t}}\}} \left|W^t_{fh,[\sum_{s=1}^{B}l_s-1]}\right|=\left|W^t_{ij,[\sum_{s=1}^{B}l_s-1]}\right|\Bigg)$$

is taken over all the possible realizations of $\bigcup_{s=1}^{B}\{d_{K'_{s-1}+1,t},...,d_{K'_{s-1}+l_s,t}\}$, which is distributed according to $\mathbf{p}$ and $\mathbf{P}$, and is equivalent to $\rho_{ij,(K_{[B]},l_{[B]})}$ as defined in (10). We also remark that, in (25d), the expected number of bits sent to any subset of users specified by $(l_{[B]})$ is calculated with respect to the first $\sum_{s=1}^{B}l_s-1$ users, i.e., users $\left[\sum_{s=1}^{B}l_s-1\right]$. That is because the number of bits of each chunk cached exclusively by any subset of $\sum_{s=1}^{B}l_s-1$ users among given $\sum_{s=1}^{B}K_s$ users is almost identical according to the law of large number. It can be concluded from (25d) that the value of $R^t_{\mathrm{MAN}}(K_{[B]})$ is irrelevant to t, given $K_{[B]}$. Thus, we simply use $R_{\mathrm{MAN}}(K_{[B]})$ in the sequel.

Given the cache content distribution $\mathbf{Q}$ and $K_{[B]}$, we have

$$\left|W^t_{ij,[\sum_{s=1}^{B}l_s-1]}\right|$$

$$= q_{ij}^{\sum_{s=1}^{B}l_s-1}(1-q_{ij})^{\sum_{s=1}^{B}K_s-\sum_{s=1}^{B}l_s+1}+o(F/B) \tag{26a}$$

$$= g_{ij,(\sum_{s=1}^{B}K_s,\sum_{s=1}^{B}l_s-1)}+o(F/B). \tag{26b}$$

Ignoring the term $o(F/B)$ and substituting $\left|W^t_{ij,[\sum_{s=1}^{B}l_s-1]}\right|$ and $\left|W_{fh,[\sum_{s=1}^{B}l_s-1]}\right|$ in (25d) yields

$$R_{\mathrm{MAN}}(K_{[B]}) = \sum\nolimits_{l_{[B]}\in[0:K_1]\times\cdots\times[0:K_B]}\prod\nolimits_{j=1}^{B}\binom{K_j}{l_j}$$
$$\sum\nolimits_{i=1}^{N}\sum\nolimits_{j=1}^{B}\rho'_{ij,(K_{[B]},\ l_{[B]})}g_{ij,(\sum_{s=1}^{B}K_s,\sum_{s=1}^{B}l_s-1)}, \tag{27}$$

where $\rho'_{ij,((K_{[B]}),\ (l_{[B]}))}$ is defined in (13). Taking the expectation over all possible realizations of $(K_{[B]})$, we obtain the average delivery rate of Algorithm 2 given as follows:

$$R_{\mathrm{MAN}} = \sum\nolimits_{k_{[B]}\in\mathcal{A}^B}\prod\nolimits_{s=1}^{B}\Pr\{K_s=k_s\}R_{\mathrm{MAN}}(k_{[B]}) \tag{28a}$$

$$= \sum\nolimits_{k_{[B]}\in\mathcal{A}^B}\prod\nolimits_{s=1}^{B}\Pr\{K_s=k_s\}$$
$$\sum\nolimits_{(l_{[B]})\in[0:k_1]\times\cdots\times[0:k_B]}\prod\nolimits_{j=1}^{B}\binom{k_j}{l_j}\cdot$$
$$\sum\nolimits_{i=1}^{N}\sum\nolimits_{j=1}^{B}\rho'_{ij,(k_{[B]},\ l_{[B]})}g_{ij,(\sum_{s=1}^{B}k_s,\sum_{s=1}^{B}l_s-1)}, \tag{28b}$$

which completes the proof.

Appendix C
Proof of Theorem 3

We first prove $\Delta\varphi_1(P_A,\mathbf{p},\mathbf{P},\mathbf{Q})$ given in (14b), which is the difference between the number of bits sent by PART 1 of Algorithm 3 and those sent by the MAN scheme for $z=1$, both averaged over all the demand combinations. We then derive $\Delta\varphi_2(P_A,\mathbf{p},\mathbf{P},\mathbf{Q})$ given in (14c), which is the difference between the number of bits sent by PART 2 of

Algorithm 3 and those delivered by the MAN scheme for $z = 2$, both averaged over all the demand realizations.

In PART 1 of Algorithm 3, the server sends the missing bits which are not cached by any user in $[K^{(t)}]$. The expected number of bits of chunk W_{ij} that are not cached by any user in $[K^{(t)}]$ is given by $F/B(1-q_{ij})^{K^{(t)}} + o(F/B)$. Recall that K_j denotes the number of users demanding the jth chunks, $j \in [B]$; i.e., $K^{(t)} = \sum_{s=1}^{B} K_s$. The probability that chunk W_{ij} is requested by at least one user at the beginning of time slot t is given by (23). By summing over $i \in [N]$ and $j \in [B]$, ignoring $o(F/B)$ term, and taking the expectation over all realizations of $K_{[B]}$, we obtain the average number of bits delivered in PART 1 of Algorithm 3 as:

$$\varphi_1 = \sum\nolimits_{k_{[B]} \in \mathcal{A}^B} \prod\nolimits_{s=1}^{B} \Pr\{K_s = k_s\} \sum\nolimits_{j=1}^{B} \sum\nolimits_{i=1}^{N} \left(1 - (1-\tilde{p}_{ij})^{k_j}\right) g_{ij,(\sum_{s=1}^{B} k_s, 0)}. \quad (29)$$

Next, we derive the average number of bits sent by Algorithm 2 for $z = 1$, denoted by $\overline{\varphi}_1$. Following the similar procedure of the proof of (28), we have

$$\overline{\varphi}_1 = \sum\nolimits_{k_{[B]} \in \mathcal{A}^B} \prod\nolimits_{s=1}^{B} \Pr\{K_s = k_s\} \sum\nolimits_{j=1}^{B} k_j \sum\nolimits_{i=1}^{N} \tilde{p}_{ij} g_{ij,(\sum_{s=1}^{B} k_s, 0)}. \quad (30)$$

Thus, we have $\Delta\varphi_1(P_A, \mathbf{p}, \mathbf{P}, \mathbf{Q}) = \overline{\varphi}_1 - \varphi_1$, which proves (14b).

Recall that the probability that chunk W_{ij} is requested by at least one user at the beginning of time slot t is given by (23). Hence, the average number of bits sent by PART 2.2 of Algorithm 3 is given by

$$\varphi_2 = \sum\nolimits_{k_{[B]} \in \mathcal{A}^B} \prod\nolimits_{s=1}^{B} \Pr\{K_s = k_s\} \sum\nolimits_{j=1}^{B} \sum\nolimits_{i=1}^{N} \left(\sum\nolimits_{s=1}^{B} k_s - 1\right) \left(1 - (1-\tilde{p}_{ij})^{k_j}\right) g_{ij,(\sum_{s=1}^{B} k_s, 1)}. \quad (31)$$

Following similar steps to the proof of (28), the number of bits sent by Algorithm 2 for $z = 2$ (or PART 2.1 of Algorithm 3) is given by

$$\overline{\varphi}_2 = \sum\nolimits_{k_{[B]} \in \mathcal{A}^B} \prod\nolimits_{s=1}^{B} \Pr\{K_s = k_s\} \left(\sum\nolimits_{j=1}^{B} \binom{k_j}{2} \cdot \sum\nolimits_{i=1}^{N} \rho'_{ij,(k_{[B]},\ (0,\ldots,l_j=2,\ldots,0))} g_{ij,(\sum_{s=1}^{B} K_s, 1)} + \sum\nolimits_{j_1=1}^{B} k_{j_1} \sum\nolimits_{j_2=j_1+1}^{B} k_{j_2} \sum\nolimits_{i=1}^{N} \sum\nolimits_{j=1}^{B} \rho'_{ij,(k_{[B]},\ (0,\ldots,l_{j_1}=1,\ldots,l_{j_2}=1,\ldots,0))} g_{ij,(\sum_{s=1}^{B} k_s, 1)} \right). \quad (32)$$

Thus, we have $\Delta\varphi_2(P_A, \mathbf{p}, \mathbf{P}, \mathbf{Q}) = \max\{\overline{\varphi}_2 - \varphi_2, 0\}$, which proves (14c), and completes the proof of Theorem 3.

Appendix D
Proof of Theorem 4

To prove Theorem 4, we first derive a lower bound on the optimal rate of any time slot t given the number of users watching different chunks, i.e., $K_1, ..., K_B$, averaged over all possible demand combinations for these users, denoted by $R_{\text{opt}}(K_{[B]}, M)$. We have

$$R_{\text{opt}}(K_{[B]}, M) \triangleq \inf \left\{ \mathbb{E}\left[R_{\mathbf{d}_t}(M) \middle| K_{[B]}, C \right] \right\}, \quad (33)$$

where the infimum is taken over all the achievable schemes, and the expectation is taken over all possible demand configurations $\mathbf{d}_t$, distributed according to $\mathbf{p}$ and $\mathbf{P}$, given $K_{[B]}$. We recall that these users are re-indexed such that users $K'_{j-1}+1, ..., K'_j$ demand the jth chunks at current time slot, for $j \in [B]$.

Inspired by [11, Appendix C], in order to lower bound $R_{\text{opt}}(K_{[B]}, M)$, we consider the following genie-aided system: we recall that $r_{1j}, ..., r_{Nj}$ is an ordered permutation of $\{\tilde{p}_{1j}, ..., \tilde{p}_{Nj}\}$, such that $r_{1j} \geq \cdots \geq r_{Nj}$, $\forall j \in [B]$. For $j \in [B]$, fix $n_j \in [N]$. As aforementioned, considering user k demanding a jth chunk, i.e., $k \in [K'_{j-1}+1 : K'_j]$, if the requested jth chunk has a normalized popularity lower than $r_{n_j j}$, i.e., $\tilde{p}_{d_{k,t}} < r_{n_j j}$, it is served by a genie at no transmission cost; otherwise, i.e., if $\tilde{p}_{d_{k,t}} \geq r_{n_j j}$, it is served by a genie at no transmission cost with probability $1 - r_{n_j j}/\tilde{p}_{d_{k,t}}$; that is, the server has to transmit the required jth chunk to this user through the shared link with probability $r_{n_j j}/\tilde{p}_{d_{k,t}}$. Thus, each user demanding a jth chunk requires service from the server, i.e., not from the genie, with probability $n_j r_{n_j j}$. This immediately implies that the total number of users who are demanding the jth chunks, and served by the server during time slot t, denoted by V_j, follows a Binomial distribution $\text{Binomial}(K_j, n_j r_{n_j j})$, i.e., $\text{V}_j \sim \text{Binomial}(K_j, n_j r_{n_j j})$.

We denote the optimal rate of the above genie-aided system by $R_{\text{genie\_opt}}(K_{[B]}, n_{[B]}, M)$. For any $n_{[B]} \in [N]^B$, it provides a lower bound on the optimal rate of the original system, i.e., $R_{\text{opt}}(K_{[B]}, M)$, since a subset of users are served by the genie. Note that, for the genie-aided system, the demands of the jth chunks that are served by the server instead of the genie are independent and uniformly distributed over all the jth chunks with a normalized popularity no less than $r_{n_j j}$, i.e., $\{W_{ij} : \tilde{p}_{ij} \geq r_{n_j j}, i \in [N]\}$, the cardinality of which is n_j according to the definition of r_{ij}, $\forall j \in [B]$. That is, for $k \in [K'_{j-1}+1 : K'_j]$,

$$\Pr(d_{k,t} = ij | \text{the } k\text{-th user requires service from server}) \triangleq \begin{cases} 1/n_j, & \text{if } \tilde{p}_{ij} \geq r_{n_j j}; \\ 0, & \text{if } \tilde{p}_{ij} < r_{n_j j}, \end{cases} \quad (34)$$

$\forall i \in [N], j \in [B]$. Let $R_{\text{opt\_unif}}(v_{[B]}, n_{[B]}, M)$ denote the optimal rate of a system including $\sum_{j=1}^{B} v_j$ users, each equipped with a cache of size MF bits, where each user in a unique subset of v_j users among them independently demands one chunk from a subset of n_j jth chunks with uniform popularity distribution, for $j \in [B]$. It follows that

$$\begin{aligned} & R_{\text{genie\_opt}}(K_{[B]}, n_{[B]}, M) \\ & \geq \mathbb{E}\left(R_{\text{opt\_unif}}(\text{V}_{[B]}, n_{[B]}, M)\right) && (35a) \\ & = \sum\nolimits_{V_{[B]} \in [K_1] \times \cdots \times [K_B]} \prod\nolimits_{j=1}^{B} \Pr(\text{V}_j = V_j) \cdot R_{\text{opt\_unif}}(V_{[B]}, n_{[B]}, M) && (35b) \\ & \geq \sum\nolimits_{V_1 = v_1}^{K_1} \cdots \sum\nolimits_{V_B = v_B}^{K_B} \prod\nolimits_{j=1}^{B} \Pr(\text{V}_j = V_j) \cdot R_{\text{opt\_unif}}(V_{[B]}, n_{[B]}, M) && (35c) \\ & \geq \prod\nolimits_{j=1}^{B} \Pr(\text{V}_j \geq v_j) R_{\text{opt\_unif}}(v_{[B]}, n_{[B]}, M), && (35d) \end{aligned}$$

where the expectation in (35a) is taken over all the values of $\mathrm{V}_{[B]}$, which yields (35b); (35c) is derived by deleting some non-negative terms; (35d) is due to the fact that the optimal rate is non-decreasing with the number of users.

In the following, we lower bound $R_{\mathrm{opt\_unif}}(v_{[B]}, n_{[B]}, M)$ by applying [12, Lemma 4].

Lemma 1. *$R_{\mathrm{opt\_unif}}(v_{[B]}, n_{[B]}, M)$ defined above should satisfy*

$$R_{\mathrm{opt\_unif}}(v_{[B]}, n_{[B]}, M) \geq \prod_{j=1}^{B} \Pr(\mathrm{Z}_j \geq z_j) R_{\mathrm{opt}}(z_{[B]}, n_{[B]}, M), \quad (36)$$

for any $z_{[B]}$, such that $z_j \in [\min\{v_j, n_j\}]$, for $j \in [B]$, where Z_j is a random variable indicating the number of distinct jth chunks requested by v_j users from a library of n_j jth chunks with a uniform popularity distribution. Furthermore, $R_{\mathrm{opt}}(z_{[B]}, n_{[B]}, M)$ is the expected rate of the optimal scheme with z_j distinct demands of the jth chunks selected uniformly at random from n_j jth chunks, for $j \in [B]$.

Below, we derive a lower bound on $R_{\mathrm{opt}}(z_{[B]}, n_{[B]}, M)$ following the cut-set technique. Since the delivery rate is non-decreasing with the number of users, we restrict to a subset of users $\mathcal{U}$ consisting of $\sum_{j=1}^{B} z_j$ users, where a distinct subset of z_j users among them request z_j distinct chunks from a subset of n_j jth chunks with uniform popularity, for $j \in [B]$. We note that there exist $\prod_{j=1}^{B} \binom{n_j}{z_j} z_j!$ demand combinations of these users, each of identical probability due to the uniform distribution of chunks. We group these demand combinations into $G_{tol} \triangleq \frac{\prod_{j=1}^{B} \binom{n_j}{z_j} z_j!}{\min_{j\in[B]} \lfloor n_j/z_j \rfloor}$ disjoint groups, denoted by $\mathfrak{G}_1, ..., \mathfrak{G}_{G_{tol}}$, such that each group consists of $\min_{j\in[B]} \lfloor n_j/z_j \rfloor$ disjoint demand combinations. Consider one such group $\mathfrak{G}_g$, $g \in [G_{tol}]$. For a demand combination in this group and a corresponding message over the shared link, say X_1^g, X_1^g and $\{Z_K^{(t)} \mid k \in \mathcal{U}\}$ allow the reconstruction of a subset of $\sum_{j=1}^{B} z_j$ chunks; similarly, for another demand combination in this group and a corresponding input to the shared link, say X_2^g, X_2^g and $\{Z_K^{(t)} \mid k \in \mathcal{U}\}$ allow the reconstruction of another disjoint subset of $\sum_{j=1}^{B} z_j$ chunks; and so on so forth. Hence, with X_1^g, ..., $X_{\min_{j\in[B]} \lfloor n_j/z_j \rfloor}^g$ and $\{Z_k^n \mid k \in \mathcal{U}\}$, each user $k \in \mathcal{U}$ can reconstruct a distinct set of $\min_{j\in[B]} \lfloor n_j/z_j \rfloor$ chunks. By considering a cut separating X_1^g, ..., $X_{\min_{j\in[B]} \lfloor n_j/z_j \rfloor}^g$ and $\{Z_k^n \mid k \in \mathcal{U}\}$ from the corresponding users, we have [24, Theorem 14.10.1]

$$\sum_{i=1}^{\min_{j\in[B]} \lfloor n_j/z_j \rfloor} |X_i^g| + \sum_{k\in\mathcal{U}} |Z_K^{(t)}| \geq \sum_{j=1}^{B} z_j \min_{j\in[B]} \lfloor n_j/z_j \rfloor. \quad (37)$$

We have

$$R_{\mathrm{opt}}(z_{[B]}, n_{[B]}, M) = \inf \left\{ \frac{1}{G_{tol}} \sum_{g=1}^{G_{tol}} \sum_{i=1}^{\min_{j\in[B]} \lfloor n_j/z_j \rfloor} \frac{|X_i^g|}{\min_{j\in[B]} \lfloor n_j/z_j \rfloor} \right\}, \quad (38)$$

where the infimum is is taken over all the achievable schemes. We also have the cache capacity constraints $MB \geq |Z_K^{(t)}|$ (normalized by F/B), for $k \in \mathcal{U}$. Plugging these into (37), we obtain

$$R_{\mathrm{opt}}(z_{[B]}, n_{[B]}, M) \geq \sum_{j=1}^{B} z_j \left(1 - \frac{MB}{\min_{j\in[B]} \lfloor n_j/z_j \rfloor}\right). \quad (39)$$

Next, we note that both V_j and Z_j are random variables expressed as self-bounding functions of random vectors (see [11, Definition 3]). We apply a concentration property of these random variables (see [11, Lemma 4]) to lower bound probabilities $\Pr(\mathrm{V}_j \geq v_j)$ and $\Pr(\mathrm{Z}_j \geq z_j)$, and find the range of v_j and z_j, for $j \in [B]$. According to [11, Lemma 4], we can write

$$\Pr\left(\mathrm{V}_j \geq \mathbb{E}\left[\mathrm{V}_j\right] - \mu\right) \geq 1 - \exp\left(-\frac{\mu^2}{2\mathbb{E}\left[\mathrm{V}_j\right]}\right), \quad (40)$$

with $0 < \mu \leq \mathbb{E}\left[\mathrm{V}_j\right]$. We have $\mathbb{E}\left[\mathrm{V}_j\right] = K_j n_j r_{n_j j}$ as $\mathrm{V}_j \sim \mathrm{Binomial}(K_j, n_j r_{n_j j})$. Letting $\mu = \mathbb{E}\left[\mathrm{V}_j\right] - v_j$, we obtain

$$\Pr\left(\mathrm{V}_j \geq v_j\right) \geq 1 - \exp\left(-\frac{(K_j n_j r_{n_j j} - v_j)^2}{2K_j n_j r_{n_j j}}\right) \triangleq f_j'(K_j, n_j, v_j), \quad (41)$$

where $0 < v_j \leq K_j n_j r_{n_j j}$, for $j \in [B]$. Similarly, we have

$$\Pr\left(\mathrm{Z}_j \geq z_j\right) \geq 1 - \exp\left(-\frac{(f(n_j, v_j) - z_j)^2}{2f(n_j, v_j)}\right) \triangleq f_j''(n_j, v_j, z_j), \quad (42)$$

where $\mathbb{E}\left[\mathrm{Z}_j\right] = n_j\left(1 - (1 - 1/n_j)^{v_j}\right) \triangleq f(n_j, v_j)$, for $0 < z_j \leq f(n_j, v_j)$, for $j \in [B]$.

Combining (35d), (36) and (39), for given $v_{[B]}$, we obtain

$$R_{\mathrm{genie\_opt}}(K_{[B]}, n_{[B]}, M) \geq \prod_{j=1}^{B} \Pr\left(\mathrm{V}_j \geq v_j\right) \max_{\substack{z_j \in [\lceil \min\{f(n_j, v_j), v_j\} \rceil] \\ j\in[B]}} \left\{ \prod_{j=1}^{B} \Pr\left(\mathrm{Z}_j \geq z_j\right) \sum_{j=1}^{B} z_j \left(1 - \frac{MB}{\min_{j\in[B]} \lfloor \frac{n_j}{z_j} \rfloor}\right) \right\}. \quad (43)$$

For any $\widetilde{z}_j \in (0, f(n_j, v_j)]$ and $j \in [B]$, we have

$$R_{\mathrm{genie\_opt}}(K_{[B]}, n_{[B]}, M) \geq \prod_{j=1}^{B} \Pr\left(\mathrm{V}_j \geq v_j\right) \max_{\substack{z_j \in [\lceil \min\{\widetilde{z}_j, v_j\} \rceil] \\ j\in[B]}} \left\{ \prod_{j=1}^{B} \Pr\left(\mathrm{Z}_j \geq z_j\right) \sum_{j=1}^{B} z_j \left(1 - \frac{MB}{\min_{j\in[B]} \lfloor \frac{n_j}{z_j} \rfloor}\right) \right\} \quad (44a)$$

$$\geq \prod_{j=1}^{B} \Pr\left(\mathrm{V}_j \geq v_j\right) \prod_{j=1}^{B} \Pr\left(\mathrm{Z}_j \geq \widetilde{z}_j\right) \quad (44b)$$

$$\max_{\substack{z_j \in [\lceil \min\{\widetilde{z}_j, v_j\} \rceil] \\ j\in[B]}} \left\{ \sum_{j=1}^{B} z_j \left(1 - \frac{MB}{\min_{j\in[B]} \lfloor \frac{n_j}{z_j} \rfloor}\right) \right\}, \quad (44c)$$

where (44a) is derived since $\widetilde{z}_j \leq f(n_j, v_j)$, $\forall j \in [B]$; (44c) follows by the fact that $z_j \leq \lceil \widetilde{z}_j \rceil$ and Z_j is an integer, $\forall j \in$

$[B]$. Using the lower bounds in (41) and (42), and optimizing over $n_{[B]}$, $v_{[B]}$, and $\widetilde{z}_{[B]}$, we have

$$R_{\text{opt}}(K_{[B]}, M) \geq \max_{n_{[B]},\ v_{[B]},\ \widetilde{z}_{[B]}} \left\{ \prod_{j=1}^{B} f_j'(K_j, n_j, v_j) \cdot \prod_{j=1}^{B} f_j''(n_j, v_j, \widetilde{z}_j) \max_{z_j \in [\lceil \min\{\widetilde{z}_j, v_j\} \rceil], j \in [B]} \left\{ \sum_{j=1}^{B} z_j \left(1 - \frac{MB}{\min_{j\in[B]} \lfloor n_j / z_j \rfloor}\right) \right\} \right\}, \tag{45}$$

where $n_j \in [N]$, $v_j \in (0, K_j n_j r_{n_j j}]$, $\widetilde{z}_j \in (0, f(n_j, v_j)]$, $j \in [B]$. Taking the expectation over all possible combinations of $K_{[B]}$, we have

$$\begin{aligned} R^*(M) &\geq \mathbb{E}[R_{\text{opt}}(K_{[B]}, M)] \\ &= \sum\nolimits_{k_{[B]} \in \mathcal{A}^B} \prod_{j=1}^{B} \Pr\{K_j = k_j\} R_{\text{opt}}(k_{[B]}, M), \end{aligned}$$

which, with (45), completes the proof of Theorem 4.

REFERENCES

[1] Q. Yang, M. Mohammadi Amiri, and D. Gündüz, "Audience retention rate aware coded video caching," in *Proc. IEEE Int'l Conf. Commun. Workshop (ICC Workshop)*, Paris, France, May 2017, pp. 1189–1194.

[2] M. Zeni, D. Miorandi, and F. De Pellegrini, "Youstatanalyzer: A tool for analysing the dynamics of youtube content popularity," in *Proc. of ICST VALUETOOLS*, Torino, Italy, Dec. 2013, pp. 286–289.

[3] N. Golrezaei, A. F. Molisch, A. G. Dimakis, and G. Caire, "Femtocaching and device-to-device collaboration: A new architecture for wireless video distribution," *IEEE Commun. Mag.*, vol. 51, no. 4, pp. 142–149, Apr. 2013.

[4] P. Blasco and D. Gunduz, "Multi-armed bandit optimization of cache content in wireless infostation networks," in *Proc. IEEE Int'l Symp. on Inform. Theory (ISIT)*, Honolulu, HI, Jun. 2014, pp. 51–55.

[5] M. Gregori, J. Gomez-Vilardebo, J. Matamoros, and D. Gündüz, "Wireless content caching for small cell and D2D networks," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 5, pp. 1222–1234, Mar 2016.

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

[7] S. O. Somuyiwa, A. György, and D. Gündüz, "A reinforcement-learning approach to proactive caching in wireless networks," *IEEE J. Sel. Areas Commun.*, to appear.

[8] M. A. Maddah-Ali and U. Niesen, "Decentralized caching attains order optimal memory-rate tradeoff," *IEEE/ACM Trans. Netw*, vol. 23, no. 4, pp. 1029–1040, Apr. 2014.

[9] M. Mohammadi Amiri, Q. Yang, and D. Gündüz, "Coded caching for a large number of users," in *Proc. IEEE Inform. Theory Workshop (ITW)*, Cambridge, UK, Sep. 2016.

[10] M. Mohammadi Amiri and D. Gündüz, "Fundamental limits of coded caching: Improved delivery rate-cache capacity trade-off," *IEEE Trans. Commun.*, vol. 65, no. 2, pp. 806–815, Feb. 2017.

[11] M. Ji, A. M. Tulino, J. Llorca, and G. Caire, "Order-optimal rate of caching and coded multicasting with random demands," *IEEE Trans. Inform. Theory*, vol. 63, no. 6, pp. 3923–3949, Jun. 2017.

[12] U. Niesen and M. A. Maddah-Ali, "Coded caching with nonuniform demands," *IEEE Trans. Inform. Theory*, vol. 63, no. 2, pp. 1146–1158, Feb. 2017.

[13] R. Pedarsani, M. A. Maddah-Ali, and U. Niesen, "Online coded caching," in *Proc. IEEE Int'l Conf. Commun. (ICC)*, Sydney, Australia, Jun. 2014, pp. 1878–1883.

[14] Q. Yang and D. Gündüz, "Coded caching and content delivery with heterogeneous distortion requirements," *IEEE Trans. Inform. Theory*, vol. 64, no. 6, pp. 4347–4364, Jun. 2018.

[15] M. Mohammadi Amiri, Q. Yang, and D. Gündüz, "Decentralized caching and coded delivery with distinct cache capacities," *IEEE Trans. Commun.*, vol. 65, no. 11, pp. 4657–4669, Nov. 2017.

[16] U. Niesen and M. A. Maddah-Ali, "Coded caching for delay-sensitive content," in *Proc. IEEE Int'l Conf. Commun. (ICC)*, London, UK, June 2015, pp. 5559–5564.

[17] H. Ghasemi and A. Ramamoorthy, "Asynchronous coded caching," in *Proc. IEEE Int'l Symp. on Inform. Theory (ISIT)*, Aachen, Germany, May 2017, pp. 2438–2442.

[18] L. Maggi, L. Gkatzikis, G. Paschos, and J. Leguay, "Adapting caching to audience retention rate: Which video chunk to store?" *Comput. Commun.*, vol. 116, pp. 159–171, Jan. 2018.

[19] E. Ozfatura and D. Gündüz, "Uncoded caching and cross-level coded delivery for non-uniform file popularity," in *Proc. IEEE Int'l Conf. Commun. (ICC)*, Kansas City, MO, May 2018.

[20] L. Wang, S. Bayhan, and J. Kangasharju, "Optimal chunking and partial caching in information-centric networks," *Comput. Commun.*, vol. 61, pp. 48–57, May 2015.

[21] L. Breslau, P. Cao, L. Fan, G. Phillips, and S. Shenker, "Web caching and Zipf-like distributions: Evidence and implications," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, NY, Mar. 1999, pp. 126–134.

[22] J. Yu, C. T. Chou, Z. Yang, X. Du, and T. Wang, "A dynamic caching algorithm based on internal popularity distribution of streaming media," *Multimedia Syst.*, vol. 12, no. 2, pp. 135–149, Jul. 2006.

[23] J. Zhang, X. Lin, and X. Wang, "Coded caching under arbitrary popularity distributions," *IEEE Trans. Inform. Theory*, vol. 64, no. 1, pp. 349–366, Jan 2018.

[24] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. New York, NY, USA: Wiley, 1991.

Qianqian Yang (S'12) received the Ph.D. degree in electrical engineering from Imperial College London in 2019. She is currently a postdoctoral research associate at Imperial College London. Her research interests include communications and information theory, signal processing and machine learning.

Mohammad Mohammadi Amiri (S'16) received the B.Sc. degree (Hons.) in Electrical Engineering from the Iran University of Science and Technology in 2011, and the M.Sc. degree (Hons.) in Electrical Engineering from the University of Tehran in 2014. He is currently pursuing the Ph.D. degree with the Imperial College London. His research interests include information and coding theory, wireless communications, machine learning, federated learning, signal processing, MIMO systems, and cooperative networks.

Deniz Gündüz (S'03-M'08-SM'13) received the M.S. and Ph.D. degrees in electrical engineering from NYU Tandon School of Engineering (formerly Polytechnic University) in 2004 and 2007, respectively. He served as a postdoctoral research associate at Princeton University, as a consulting assistant professor at Stanford University, and a research associate at CTTC in Barcelona, Spain, before joining the Electrical and Electronic Engineering Department of Imperial College London, UK, where he is currently a Reader (Associate Professor) in information theory and communications, and leads the Information Processing and Communications Laboratory (IPC-Lab). His research interests lie in the areas of communications and information theory, machine learning, and privacy.