

# Centralized Caching and Delivery of Correlated Contents over a Gaussian Broadcast Channel

Qianqian Yang\*, Parisa Hassanzadeh†, Deniz Gündüz\*, Elza Erkip†

\*Electrical and Electronic Engineering Department, Imperial College London, London SW7 2AZ, U.K.

{q.yang14, d.gunduz}@imperial.ac.uk

† Electrical and Computer Engineering Department, New York University, Brooklyn, NY

{ph990, elza}@nyu.edu

**Abstract**—Content delivery in a multi-user cache-aided broadcast network is studied, where a server holding a database of correlated contents communicates with the users over a Gaussian broadcast channel (BC). The minimum transmission power required to satisfy all possible demand combinations is studied, when the users are equipped with caches of equal size. A lower bound on the required transmit power is derived, assuming uncoded cache placement, as a function of the cache capacity. A centralized joint cache and channel coding scheme is proposed, which not only utilizes the user’s local caches, but also exploits the correlation among the contents in the database. This scheme provides an upper bound on the minimum required transmit power for a given cache capacity. Our results indicate that exploiting the correlations among the contents in a cache-aided Gaussian BC can provide significant energy savings.

## I. INTRODUCTION

Thanks to the decreasing cost and increasing capacity of memory available at mobile devices, *proactive caching* has been considered as a low-cost and effective solution to today’s exponentially growing mobile data traffic [1]–[3]. Proactively storing popular contents in cache memories distributed across the network during off-peak traffic periods can greatly reduce both the network congestion and the latency during peak traffic hours. *Coded caching* [2] further exploits the broadcasting nature of wireless delivery and the contents available in users’ local cache memories to create multicasting opportunities, even when the users request distinct files, which further boosts the benefits of caching. The significant gains of coded caching over traditional uncoded caching schemes have inspired numerous studies, among which [4]–[15] are most related to this paper.

Delivering correlated contents over an error-free shared cache-aided link is considered in [4]–[9]. In [4], correlation among an arbitrary number of files in a caching system is exploited by identifying the most representative files, which are then used as references for compressing the remaining correlated files. A two-file system is studied in detail in [7], where the files are initially compressed using Gray-Wyner source coding, and an optimal caching scheme is derived for the two-receiver network. This scheme is generalized to more files in [6], which is shown to be optimal for large cache

sizes. Arbitrary numbers of users and files are considered in [5], where each file is modeled as a collection of subfiles, each of which is shared by a different subset of files. In [9] a caching scheme is proposed, which exploits correlation by performing on-demand compression during delivery, resulting in a scheme that is robust to dynamic changes in a correlated library. Cache-aided content delivery is studied in [16] with a source coding perspective, where users request different quality lossy versions of the files in the library. More recently, [11]–[15], and others have relaxed the assumption of a shared error-free bit-pipe from the server to the users by replacing it with noisy BCs in the traditional setting of uncorrelated files. In [14] the authors consider a total memory budget, and optimize the cache assignment with respect to the channel capacity for degraded BCs. In [12] the benefits of caching and coded delivery are studied from the perspective of energy-efficiency in Gaussian BCs.

In this paper, we consider the degraded Gaussian BC model studied in [12], but we assume that the files in the library can be arbitrarily correlated as modeled in [5]. We evaluate the performance in terms of the minimum transmission power required to satisfy any demand combination, called the *peak power* in [12] as it represents the power required to satisfy the worst demand combination. We derive a lower bound on the minimum required transmission power assuming uncoded cache placement, and an upper bound that employs superposition coding and power allocation when delivering coded contents over the BC. Through numerical simulations, we show that the proposed correlation-aware joint caching and channel coding scheme reduces the transmission power significantly compared to schemes that are correlation-ignorant.

*Notations:*  $\mathbb{R}$  is the set of all the real values, while  $\mathbb{R}^+$  denotes the set of all the positive real values. The set of integers  $\{i, \dots, j\}$ , where  $i \leq j$ , is denoted by  $[i : j]$ , and for  $q \in \mathbb{R}^+$ , the set  $[1 : \lceil q \rceil]$  is denoted shortly by  $[q]$ . For sets  $\mathcal{A}$  and  $\mathcal{B}$ , we define  $\mathcal{A} \setminus \mathcal{B} \triangleq \{x : x \in \mathcal{A}, x \notin \mathcal{B}\}$ , and  $|\mathcal{A}|$  denotes the cardinality of  $\mathcal{A}$ .  $\binom{j}{i}$  represents the binomial coefficient if  $j \geq i$ ; otherwise,  $\binom{j}{i} = 0$ . For event  $E$ ,  $\mathbb{1}\{E\} = 1$  if  $E$  is true; and  $\mathbb{1}\{E\} = 0$ , otherwise.

## II. SYSTEM MODEL

Consider a wireless server that holds a database of  $N$  correlated files, denoted by  $\mathbf{W} = (W_1, \dots, W_N)$ , which have

This work has been supported by NSF Grant #1619129, and by the European Research Council (ERC) Staring Grant BEACON (grant agreement No. 725731).

been jointly compressed into independent subfiles. File  $W_i$ ,  $i \in [N]$ , consists of  $2^{N-1}$  independent subfiles, i.e.,

$$W_i = \{\overline{W}_S : S \subseteq [N], i \in S\}, \quad (1)$$

where  $\overline{W}_S$  denotes the subfile that is shared exclusively by the files  $\{W_i : i \in S\}$ . The subfiles are arranged into  $N$  *sublibraries*,  $L_1, \dots, L_N$ , such that  $L_\ell$  contains all the subfiles that are shared exclusively by a set of  $\ell$  files, i.e.,

$$L_\ell = \{\overline{W}_S : S \subseteq [N], |S| = \ell\}. \quad (2)$$

Motivated by Gray-Wyner compression, we assume that all the subfiles in a sublibrary have the same length, and subfile  $\overline{W}_S \in L_\ell$  is distributed uniformly over the set  $[2^{nR_\ell}]$ , where  $R_\ell$  is the rate of subfile  $\overline{W}_S$  in bits per channel use, and  $n$  denotes the transmission blocklength, referring to  $n$  uses of the BC. We define  $\mathbf{R} \triangleq (R_1, \dots, R_N)$ . All the files are of the same normalized rate of  $R$  bits per channel use, where

$$R = \sum_{\ell=1}^N \binom{N-1}{\ell-1} R_\ell. \quad (3)$$

Each user is equipped with a cache of size  $nM$  bits, where  $M$  is called the *normalized cache capacity*. Communication takes place in two phases. During the first phase, referred to as the *placement phase*, the user caches are filled by the server without the knowledge of particular demands. This phase happens during a period of low traffic, which can be considered noiseless. In this paper, we are restricted to *uncoded cache placement phase*, i.e, users only caches uncoded messages. We consider *centralized caching*; that is, the server has the knowledge of the active users in advance, allowing the cache placement to be conducted in a coordinated fashion. At the beginning of the second phase, referred to as the *delivery phase*, user  $k \in [K]$  requests file  $W_{d_k}$  from the library, with  $d_k$  uniformly distributed over  $[N]$ . Let  $\mathbf{d} \triangleq (d_1, \dots, d_K)$  denote the demand vector. All the requests are satisfied through a Gaussian BC, characterized by a time-invariant channel vector  $\mathbf{h} = (h_1, \dots, h_K)$  and additive white Gaussian noise, where  $h_k$  denotes the real channel gain between the server and user  $k$ . The channel gains are fixed, and are known to all the parties. Without loss of generality, we assume  $h_1^2 \leq h_2^2 \leq \dots \leq h_K^2$ , such that the users are ordered from the weakest to the strongest. The  $i^{\text{th}}$  channel output, is given by

$$Y_{k,i} = h_k X_i + \sigma_{k,i}, \quad (4)$$

for channel gain  $h_k$ , and noise  $\sigma_{k,i} \sim \mathcal{N}(0, 1)$  at user  $k$  in the  $i^{\text{th}}$  channel use, which is assumed to be independently and identically distributed across time and users.

An  $(n, \mathbf{R}, M, P)$  code for this system consists of:

- **$K$  caching functions**  $f_k$ ,  $k \in [K]$ ,

$$f_k : [2^{nR}]^N \times \mathbb{R}^K \rightarrow [2^{nM}], \quad (5)$$

and user  $k$ 's cache content, denoted by  $Z_k$ , is given by  $Z_k = f_k(\mathbf{W}, \mathbf{h})$ . Let  $\mathbf{Z} \triangleq (Z_1, \dots, Z_K)$

- **A delivery function**  $g$ ,

$$g : [2^{nR}]^N \times [2^{nM}]^K \times \mathbb{R}^K \times [N]^K \rightarrow \mathbb{R}^n, \quad (6)$$

which, for given cache contents  $\mathbf{Z}$ , channel gains  $\mathbf{h}$ , and demand vector  $\mathbf{d}$ , generates the channel input signal,  $X^n(\mathbf{W}, \mathbf{d}) = g(\mathbf{W}, \mathbf{Z}, \mathbf{h}, \mathbf{d})$ , transmitted over the Gaussian BC in  $n$  channel uses, with  $X_i(\mathbf{W}, \mathbf{d})$  denoting the  $i^{\text{th}}$  channel input. The channel input vector is generated such that its average power is not more than  $P$ , i.e.,

$$P(\mathbf{W}, \mathbf{d}) \triangleq \sum_{i=1}^n X_i^2(\mathbf{W}, \mathbf{d}) \leq P, \quad \forall \mathbf{d} \in [N]^K. \quad (7)$$

- **$K$  decoding functions**  $\phi_k$ ,  $k \in [K]$ ,

$$\phi_k : \mathbb{R}^n \times [2^{nM}] \times \mathbb{R}^K \times [N]^K \rightarrow [2^{nR}], \quad (8)$$

where  $\widehat{W}_{d_k} = \phi_k(Y^n, Z_k, \mathbf{h}, \mathbf{d})$ , is the reconstruction of  $W_{d_k}$ , and  $Y^n$  is the channel output at user  $k$ .

**Definition 1.** A cache capacity-power pair  $(M, P)$  is achievable for the system described above, if there exists a sequence of  $(n, \mathbf{R}, M, P)$  codes such that

$$\lim_{n \rightarrow \infty} \mathbb{P} \left\{ \bigcup_{\mathbf{d} \in [N]^K} \bigcup_{k=1}^K \{ \widehat{W}_{d_k} \neq W_{d_k} \} \right\} = 0. \quad (9)$$

For a given cache capacity  $M$ , our goal is to characterize the minimum power  $P$  such that  $(M, P)$  is achievable. The corresponding *memory-power function* is defined as

$$P^*(M) \triangleq \inf \{ P : (M, P) \text{ is achievable} \}. \quad (10)$$

We conclude the section with the following proposition, which is frequently referred to in the remainder of this paper.

**Proposition 1.** [17] In a  $K$ -user degraded Gaussian BC with  $h_1^2 \leq h_2^2 \leq \dots \leq h_K^2$ ,  $K$  distinct messages with rates  $\rho_1, \dots, \rho_K$ , can be reliably transmitted to users  $1, \dots, K$ , respectively, iff

$$\rho_k \leq C \left( \frac{h_k^2 P_k}{1 + h_k^2 \sum_{j=k+1}^K P_j} \right), \quad k = 1, \dots, K, \quad (11)$$

for some  $P_1, \dots, P_K$ , which can be achieved by using superposition coding with Gaussian codewords of power  $P_1, \dots, P_K$ , where  $C(x) \triangleq \frac{1}{2} \log_2(1 + x)$ . The minimum total transmit power for reliable communication is thus given by

$$P = \sum_{k=1}^K P_k \geq \sum_{k=1}^K \left( \frac{2^{2\rho_k} - 1}{h_k^2} \right) \prod_{j=1}^{k-1} \frac{2^{2\rho_j}}{h_j^2}. \quad (12)$$

### III. MAIN RESULTS

This section provides a lower and an upper bounds on the memory-power function,  $P^*(M)$ .

#### A. Lower Bound on $P^*(M)$

The following lower bound is stated here without a proof, which will be provided in a longer version of this paper.

**Theorem 1.** For the caching problem described in Section II with uncoded cache placement phase, the optimal memory-power function,  $P^*(M)$ , is lower bounded as

$$P^*(M) \geq P_{LB}(M) \triangleq \sum_{k=1}^{\min\{N, K\}} \left( \frac{2^{2\bar{\rho}_k} - 1}{h_k^2} \right) \prod_{j=1}^{k-1} \frac{2^{2\bar{\rho}_j}}{h_j^2}, \quad (13)$$

where

$$\tilde{\rho}_k \triangleq \max \left\{ \sum_{\ell=0}^{N-k} \binom{N-k}{\ell} R_{\ell+1} - M, 0 \right\}, \forall k \in [K]. \quad (14)$$

### B. Upper Bound on $P^*(M)$

**Theorem 2.** For the caching problem described in Section II, the optimal memory-power function,  $P^*(M)$ , is upper bounded as

$$\begin{aligned} P^*(M) &\leq \min_{\boldsymbol{\pi}=(\pi_1, \dots, \pi_N)} P_{UB}(M, \boldsymbol{\pi}), \\ \text{s.t.} \quad &\sum_{i=1}^N \pi_i \leq 1, \\ &0 \leq \pi_i \leq 1, \quad i = 1, \dots, N, \end{aligned}$$

where

$$P_{UB}(M, \boldsymbol{\pi}) \triangleq \sum_{k=1}^{\min\{N, K\}} \left( \frac{2^{2\tilde{\rho}_k} - 1}{h_k^2} \right) \prod_{j=1}^{k-1} 2^{2\tilde{\rho}_j}, \quad (15a)$$

$$\hat{\rho}_k \triangleq \sum_{\ell=1}^N \sum_{r=\max\{\ell-N+K, 1\}}^{\min\{\ell, K\}} \binom{N-K}{\ell-r} \binom{\min\{N, K\}-1}{r-1} \gamma_{k, \ell, r} \quad (15b)$$

$$\begin{aligned} \gamma_{k, \ell, r} &\triangleq \\ &\begin{cases} \left( \frac{\lfloor t_\ell \rfloor - t_\ell + 1}{\lfloor t_\ell \rfloor - t_\ell} \lambda(\lfloor t_\ell \rfloor) - \lambda(\lfloor t_\ell \rfloor + 1) \right) R_\ell, & \text{if } k \in \left[ \frac{\min\{N, K\}}{r} + 1 \right] \\ 0, & \text{otherwise} \end{cases} \\ \lambda(t) &\triangleq \frac{\binom{K-k}{\lfloor t \rfloor}}{\binom{K}{\lfloor t \rfloor}} (\lfloor t \rfloor - t), \end{aligned} \quad (15c)$$

$$t_\ell \triangleq \frac{K\pi_\ell M}{\binom{N}{\ell} R_\ell}. \quad (15d)$$

*Proof.* The proof can be derived by characterizing the transmit power achieved by the coding scheme outlined in Section IV, and will be provided in the longer version.  $\square$

## IV. CACHE-AIDED SUPERPOSITION CODING

The upper bound in Theorem 2 is achieved by a centralized caching and delivery scheme, which optimizes the cache allocation among different libraries and places uncoded messages into users' caches in a centralized manner. Superposition coding is employed to losslessly deliver coded messages over the Gaussian BC [17], where the coded messages are generated taking into account the correlation among the requested files as well as the channel gains. As in [5]–[7], the scheme operates by treating the sublibraries independent of during the placement and delivery phases to determine the cache content and messages targeted at each user, which are then jointly delivered over the BC. In the following, we first illustrate the main idea through a simple example, and then provide the general description of the coding scheme.

### A. Motivating Example

Consider a system with  $K = 3$  users having channel gains  $h_1^2 \leq h_2^2 \leq h_3^2$ , and a database of  $N = 3$  files with sublibraries:

- $L_1 = \{\overline{W}_{\{1\}}, \overline{W}_{\{2\}}, \overline{W}_{\{3\}}\}$ , each with rate  $R_1$ .
- $L_2 = \{\overline{W}_{\{1,2\}}, \overline{W}_{\{2,3\}}, \overline{W}_{\{1,3\}}\}$ , each with rate  $R_2$ .
- $L_3 = \{\overline{W}_{\{1,2,3\}}\}$ , with rate  $R_3$ .

Each user has a cache with capacity  $M = R_1 + R_2 + \frac{1}{3}R_3$ .

◦ **Placement Phase:** Since content placement is carried out independently across the sublibraries, for ease of exposition, we assume that the cache capacity is divided into three portions, such that the users allocate portions with normalized capacities  $R_1$ ,  $R_2$ , and  $\frac{1}{3}R_3$  for caching files from sublibraries  $L_1$ ,  $L_2$  and  $L_3$ , respectively. We remark that the cache capacity allocation in this example is not optimal, and the proposed scheme further optimizes the allocation as described in more detail in Subsection IV-B. We use the prefetching policy proposed in [10], which divides the subfiles in sublibrary  $L_\ell$  into three non-overlapping packets of size  $\frac{1}{3}nR_\ell$  bits. Then, user  $k$  caches

$$Z_k = \left\{ \overline{W}_{\{1\},\{k\}}, \overline{W}_{\{2\},\{k\}}, \overline{W}_{\{3\},\{k\}}, \overline{W}_{\{1,2\},\{k\}}, \overline{W}_{\{2,3\},\{k\}}, \overline{W}_{\{1,3\},\{k\}}, \overline{W}_{\{1,2,3\},\{k\}} \right\},$$

where  $\overline{W}_{S,\{k\}}$  denotes the  $k^{\text{th}}$  packet of subfile  $\overline{W}_S$  cached at user  $k \in [3]$ .

◦ **Delivery Phase:** During the delivery phase, once the demand vector is revealed, the server computes the messages intended for each user, independently from each sublibrary, and then delivers them over the BC via superposition coding. Consider demand vector  $\mathbf{d} = (1, 2, 3)$ . User 1, who is the weakest user, requires all parts of subfiles  $\{\overline{W}_{\{1\}}, \overline{W}_{\{1,2\}}, \overline{W}_{\{1,3\}}, \overline{W}_{\{1,2,3\}}\}$  that are missing from its cache, in order to reconstruct  $W_1$ . User 2 requires the four subfiles corresponding to file  $W_2$ , but having a better channel compared to user 1, through successive cancellation, it can also decode the messages targeted at user 1. Similarly, user 3 can decode both messages intended for the weaker users. User messages from each sublibrary are determined as follows.

- Sublibrary  $L_1$ : Based on the demand, all subfiles in  $L_1$  are required by the users. User 1 needs to receive  $\overline{W}_{\{1\},\{2\}}$  and  $\overline{W}_{\{1\},\{3\}}$ , whose targeted message, denoted by  $V_{\mathbf{d}}^1(L_1)$ , is generated as follows:

$$V_{\mathbf{d}}^1(L_1) = \{\overline{W}_{\{1\},\{2\}} \oplus \overline{W}_{\{2\},\{1\}}, \overline{W}_{\{1\},\{3\}} \oplus \overline{W}_{\{3\},\{1\}}\}. \quad (16)$$

Since user 2 is able to decode its required packet  $\overline{W}_{\{2\},\{1\}}$  from message  $V_{\mathbf{d}}^1(L_1)$ , it only needs  $\overline{W}_{\{2\},\{3\}}$ , which is recovered through the message

$$V_{\mathbf{d}}^2(L_1) = \{\overline{W}_{\{2\},\{3\}} \oplus \overline{W}_{\{3\},\{2\}}\}. \quad (17)$$

User 3 can decode its missing packets from  $V_{\mathbf{d}}^1(L_1)$  and  $V_{\mathbf{d}}^2(L_1)$ , and therefore,  $V_{\mathbf{d}}^3(L_1) = \emptyset$ .

- Sublibrary  $L_2$ : Each user requires two subfiles from  $L_2$ , which can be considered as two separate demands. One possible partitioning could be  $\mathfrak{S}_1 = (\{1, 2\}, \{1, 2\}, \{1, 3\})$  and  $\mathfrak{S}_2 = (\{1, 3\}, \{2, 3\}, \{2, 3\})$ , where  $\mathfrak{S}_1$  corresponds to users 1, 2 and 3 requesting subfiles  $\overline{W}_{\{1,2\}}$ ,  $\overline{W}_{\{1,2\}}$  and  $\overline{W}_{\{1,3\}}$ , respectively. Then  $V_{\mathbf{d}}^k(L_2) = \{v_1^k, v_2^k\}$ , where  $v_i^k$  is user  $k$ 's message corresponding to demand  $\mathfrak{S}_i$ . Then, for  $\mathfrak{S}_1$

$$v_1^1 = \{\overline{W}_{\{1,2\},\{2\}} \oplus \overline{W}_{\{1,2\},\{1\}}, \overline{W}_{\{1,2\},\{3\}} \oplus \overline{W}_{\{1,3\},\{1\}}\}, \quad (18)$$

$$v_2^1 = \{\overline{W}_{\{1,3\},\{2\}} \oplus \overline{W}_{\{1,2\},\{3\}}\}, \quad (19)$$

$$v_3^1 = \emptyset, \quad (20)$$

and for  $\mathfrak{S}_2$

$$v_2^1 = \{\overline{W}_{\{1,3\},\{2\}} \oplus \overline{W}_{\{2,3\},\{1\}}, \overline{W}_{\{1,3\},\{3\}} \oplus \overline{W}_{\{2,3\},\{1\}}\}, \quad (21)$$

$$v_2^2 = \{\overline{W}_{\{2,3\},\{2\}} \oplus \overline{W}_{\{2,3\},\{3\}}\}, \quad (22)$$

$$v_2^3 = \emptyset. \quad (23)$$

- Sublibrary  $L_3$ : All users require  $\overline{W}_{\{1,2,3\}}$ , and therefore

$$V_d^1(L_3) = \{\overline{W}_{\{1,2,3\},\{2\}} \oplus \overline{W}_{\{1,2,3\},\{1\}}, \overline{W}_{\{1,2,3\},\{3\}} \oplus \overline{W}_{\{1,2,3\},\{1\}}\}, \quad (24)$$

$$V_d^2(L_2) = V_d^3(L_2) = \emptyset. \quad (25)$$

The messages in (16), (18), (21) and (24) constitute all the messages targeted at user 1, with total rate  $\rho_1 = 2(R_1 + 2R_2 + R_3)$ . Messages (17), (20) and (22) are targeted at user 2 with total rate  $\rho_2 = R_1 + 2R_2$ , and finally, user 3 can successfully recover its requested file from the messages intended for users 1 and 2, i.e.,  $\rho_3 = 0$ . Based on Proposition 1, the target rates can be delivered to the users with superposition coding of Gaussian codewords satisfying (11), with a minimum power value given in (12).

### B. Proposed Scheme

This section presents the description of the proposed centralized caching and delivery scheme that generalizes the example in Subsection IV-A, and achieves the transmission power stated in Theorem 2. Similarly to the schemes in [5]–[7], the proposed scheme treats the sublibraries independently: 1) the cache capacity is optimally divided among the  $N$  sublibraries, and user caches are filled in a centralized fashion, 2) for each demand realization, the server identifies the messages that need to be delivered to each user, independently across sublibraries, using a modified version of the scheme proposed in [5], and 3) the server employs superposition coding to reliably communicate coded messages over the Gaussian BC.

1) **Placement Phase:** As in [5]–[7], the contents placed in each user's cache are identified separately for the different sublibraries, each with a different level of commonness. To this end, a fraction of the available cache capacity is allocated to each sublibrary. Let  $\boldsymbol{\pi} = (\pi_1, \dots, \pi_N)$  denote the cache allocation vector, where  $\pi_\ell \in [0, 1]$  denotes the fraction of  $M$  allocated to sublibrary  $L_\ell$ , with  $\sum_{\ell=1}^N \pi_\ell = 1$ . We will later optimize  $\boldsymbol{\pi}$  to minimize the required total transmit power. For a given  $\boldsymbol{\pi}$ , placement for sublibrary  $L_\ell$  is carried out using the prefetching scheme proposed in [10] as follows. Let

$$t_\ell \triangleq \frac{K\pi_\ell M}{\binom{N}{\ell} R_\ell}, \quad t_\ell \in [0, K], \quad (26)$$

which, differently from [10], is not necessarily an integer. We address this by memory-sharing among the neighboring integer points,  $t_\ell^A \triangleq \lfloor t_\ell \rfloor$  and  $t_\ell^B \triangleq \lfloor t_\ell \rfloor + 1$ , and divide each subfile  $\overline{W}_S \in L_\ell$  into two non-overlapping parts. More specifically,  $\overline{W}_S^A = (\overline{W}_S^A, \overline{W}_S^B)$ , where  $\overline{W}_S^A$  is at rate  $(t_\ell^B - t_\ell)R_\ell$ , while  $\overline{W}_S^B$  is at rate  $(t_\ell - t_\ell^A)R_\ell$ . The prefetching policy of [10] is implemented separately for  $\{\overline{W}_S^A : S \in L_\ell\}$

and  $\{\overline{W}_S^B : S \in L_\ell\}$ . Each part  $\overline{W}_S^A$  is split into  $\binom{K}{t_\ell^A}$  non-overlapping equal-length packets, with size  $n(t_\ell^B - t_\ell) / \binom{K}{t_\ell^A} R_\ell$  bits. These packets are assigned to sets  $\mathcal{A} \subseteq [K]$  with size  $t_\ell^A$ . We denote the packet assigned to  $\mathcal{A}$  by  $\overline{W}_{S,\mathcal{A}}^A$ ; therefore,

$$\overline{W}_S^A = \{\overline{W}_{S,\mathcal{A}}^A : \mathcal{A} \subseteq [K], |\mathcal{A}| = t_\ell^A\}. \quad (27)$$

Similarly, each part  $\overline{W}_S^B$  is split into  $\binom{K}{t_\ell^B}$  non-overlapping equal-length packets, which are labeled as

$$\overline{W}_S^B = \{\overline{W}_{S,\mathcal{B}}^B : \mathcal{B} \subseteq [K], |\mathcal{B}| = t_\ell^B\}. \quad (28)$$

Given this packetization, user  $k$  caches packets  $\overline{W}_{S,\mathcal{A}}^A$  if  $k \in \mathcal{A}$ , and packets  $\overline{W}_{S,\mathcal{B}}^B$  if  $k \in \mathcal{B}$ . With this placement strategy, for each subfile in sublibrary  $L_\ell$ ,  $\binom{K-1}{t_\ell^A-1}$  distinct packets from  $\overline{W}_S^A$ , and  $\binom{K-1}{t_\ell^B-1}$  distinct packets from  $\overline{W}_S^B$ , are placed in each user's cache, amounting to a total of  $nt_\ell R_\ell / K$  bits, which satisfies the capacity constraint of  $n\pi_\ell M$  bits.

2) **Delivery Phase:** As explained in [5]–[7], delivering a file from a correlated library, which has been compressed into multiple subfiles, is a multiple-demand problem. For a given demand vector  $\mathbf{d}$ , user  $k$  requires  $\binom{N-1}{\ell-1}$  subfiles from sublibrary  $L_\ell$ , in order to successfully reconstruct file  $W_{d_k}$ . Since the sublibraries are treated independently, message  $V_d^k$ , targeted at user  $k$ , constitutes the messages computed from all the sublibraries, i.e.,

$$V_d^k = \bigcup_{\ell=1}^N V_d^k(L_\ell), \quad (29)$$

where  $V_d^k(L_\ell)$  denotes the set of messages from sublibrary  $L_\ell$  targeted at user  $k$ . Messages  $V_d^1(L_\ell), \dots, V_d^K(L_\ell)$  are determined using Algorithm 1, which is based on [5, Algorithms 1, 2]. The main idea is to treat subfiles  $\{\overline{W}_S : d_k \in \mathcal{S}\}$  that are not cached at user  $k$ , as different demands. The algorithm operates by partitioning all the requested subfiles from sublibrary  $L_\ell$  into groups, such that each user requires at most one subfile in each group; thus, resulting in a single-demand problem.

For sublibrary  $L_\ell$ ,  $V_d^1(L_\ell), \dots, V_d^K(L_\ell)$ , are generated as follows:

- Grouping the requested subfiles:

Let  $\mathcal{D} \triangleq \{d_1, \dots, d_K\}$  denote the set of distinct files requested in demand vector  $\mathbf{d}$ . The subfiles that need to be delivered to at least  $\ell$  users, are given by:

$$\{\overline{W}_S : \overline{W}_S \in L_\ell, \mathcal{S} \subseteq \mathcal{D}\}. \quad (30)$$

Since each user can request multiple subfiles from (30), they are grouped into multiple (possibly overlapping) sets with minimum cardinality, such that each group represents the demand set of a single-demand network with  $K$  users. The grouping process tries to minimize the number of distinct demands within each corresponding single-demand network. For sublibrary  $L_\ell$ , where each subfile is required by  $\ell$  distinct demands, there are at most  $\lceil |\mathcal{D}| / \ell \rceil + 1$  subfiles in each group. Note that, the subfiles in (30) are not the only contents that need to

be delivered from sublibrary  $L_\ell$ . Based on the demand vector, any subfile  $\overline{W}_S$  whose index  $S$  includes at least one of the indices in  $\mathcal{D}$ , i.e.,  $S \cap \mathcal{D} \neq \emptyset$ , is required for the lossless reconstruction of the corresponding requested file in  $\mathcal{D}$ . All such subfiles need to be identified, and grouped in a similar fashion. The subfiles in (30) correspond to  $|\mathcal{S} \cap \mathcal{D}| = \ell$ . For  $r = 1, \dots, \ell$ , we define the requested subfiles  $\mathcal{W}_r$ , as

$$\mathcal{W}_r \triangleq \{\overline{W}_S : |S| = \ell, |\mathcal{S} \cap \mathcal{D}| = r\}. \quad (31)$$

Then, each set  $\mathcal{W}_r$  is grouped using the function GROUP in Algorithm 1, which assigns a demand vector  $\mathfrak{S} = (\mathcal{S}_1, \dots, \mathcal{S}_K)$  to each group, resulting in a single-demand network with  $K$  users, where user  $k$  requests subfile  $\overline{W}_{\mathcal{S}_k}$ .

ii) Delivering the demands corresponding to each group:

The groups that have been formed above are treated independently in the delivery phase. More specifically, for a group with corresponding demand vector  $\mathfrak{S}$ , function SINGLE-DEMAND in Algorithm 1 identifies the messages  $V^1, \dots, V^K$  that need to be transmitted so that all the users recover their requested subfiles in  $\mathfrak{S}$ . These messages are computed using the scheme in [10], applied to a degraded BC. The channel is taken into account by selecting the *weakest* users with distinct demands as *leaders*, i.e., the demand of a leader is not requested by any of the weaker users,  $\{k : \mathcal{S}_k \notin \{\mathcal{S}_1, \dots, \mathcal{S}_{k-1}\}\}$ , and then greedily broadcasting XORed messages that benefit at least one leader. Note that choosing the weakest user, among users requiring the same subfile  $\overline{W}_S$ , as the leader, allows all the stronger users to decode the subfile through superposition coding and successive cancellation decoding. As mentioned previously, the proposed scheme uses memory-sharing to cache and deliver the subfiles in  $L_\ell$ , for the two parts  $\overline{W}_S^A$  and  $\overline{W}_S^B$ ; and therefore, function SINGLE-DEMAND is executed for both parts.

---

**Algorithm 1** Generate messages  $\{V_d^1(L_\ell), \dots, V_d^K(L_\ell)\}$

---

```

1:  $V_d^k(L_\ell) \leftarrow \emptyset, \forall k \in \{1, \dots, K\}$ 
2: for  $r \in \{1, \dots, \ell\}$  do
3:    $\mathcal{W}_r = \{\overline{W}_S : |S| = \ell, |\mathcal{S} \cap \mathcal{D}| = r\}$ 
4:    $\mathfrak{S}_1, \dots, \mathfrak{S}_g \leftarrow \text{Group}(\mathcal{W}_r, \mathcal{D}, \ell, r)$ 
5:   for  $i \in \{1, \dots, g\}$  do
6:      $V_A^1, \dots, V_A^K \leftarrow \text{Single-Demand}(A, \mathfrak{S}_i, t_\ell^A)$ 
7:      $V_B^1, \dots, V_B^K \leftarrow \text{Single-Demand}(B, \mathfrak{S}_i, t_\ell^B)$ 
8:      $V_d^k(L_\ell) \leftarrow V_d^k(L_\ell) \cup \{V_A^k, V_B^k\}, \forall k \in \{1, \dots, K\}$ 
9:   end for
10: end for

```

---

Message  $V_d^k(L_\ell)$  targeted at user  $k$  is the union of all the messages for sublibrary  $L_\ell$  computed for each group identified from the subfile sets  $\{\mathcal{W}_1, \dots, \mathcal{W}_\ell\}$ . The overall message for user  $k$ ,  $V_d^k$ , is thus obtained by (29). For a given demand vector  $\mathbf{d}$ , messages  $V_d^1, \dots, V_d^K$  can be reliably transmitted to users  $1, \dots, K$ , using a  $K$ -level Gaussian superposition codebook [17]. The  $k^{\text{th}}$ -level codebook consists of  $2^{n\rho_k}$  codewords,

where  $\rho_k$  is the total rate of the messages in  $V_d^k$ . The total required transmit power is given by (11) in Proposition 1.

---

```

1: function GROUP ( $\mathcal{W}, \mathcal{D}, \ell, r$ )
2: Output: Group demands  $\mathfrak{S}_1, \dots, \mathfrak{S}_g$ 
3:    $\mathcal{F} \leftarrow \mathcal{D}, \overline{\mathcal{F}} \leftarrow \emptyset, \overline{\mathcal{S}} \leftarrow \emptyset, g = 0$ 
4:   while  $\mathcal{W} \neq \emptyset$  do
5:     while  $\mathcal{F} \neq \emptyset$  do
6:       if  $|\mathcal{F}| \geq r$  then
7:         if  $\overline{\mathcal{F}} = \emptyset$  then
8:           Randomly pick  $\overline{W}_S \in \mathcal{W}$  such that  $S \cap \mathcal{D} \subseteq \mathcal{F}$ 
9:            $\mathcal{W} \leftarrow \mathcal{W} / \overline{W}_S, \mathcal{F} \leftarrow \mathcal{F} \setminus S$ 
10:          for  $d_k \in S \cap \mathcal{D}$  do
11:             $\mathcal{S}_k \leftarrow S$ 
12:          end for
13:          else
14:            for  $d_k \in \overline{\mathcal{F}}$  do
15:               $\mathcal{S}_k \leftarrow \overline{\mathcal{F}}$ 
16:            end for
17:             $\mathcal{F} \leftarrow \mathcal{F} \setminus \overline{\mathcal{F}}, \overline{\mathcal{S}} \leftarrow \emptyset, \overline{\mathcal{F}} \leftarrow \emptyset,$ 
18:            end if
19:          else
20:            Randomly pick  $\overline{W}_S \in \mathcal{W}$  such that  $\mathcal{F} \subseteq S$ 
21:            for  $d_k \in \mathcal{F}$  do
22:               $\mathcal{S}_k \leftarrow S$ 
23:            end for
24:             $\mathcal{F} \leftarrow \emptyset, \overline{\mathcal{S}} \leftarrow S, \overline{\mathcal{F}} \leftarrow S \setminus \mathcal{F}$ 
25:            end if
26:          end while
27:           $g = g + 1$ 
28:           $\mathfrak{S}_g = (\mathcal{S}_1, \dots, \mathcal{S}_K)$ 
29:        end while
30: end function

```

---



---

```

1: function SINGLE-DEMAND ( $C, \mathfrak{S}, t$ )
2: Input:  $\mathfrak{S} = (\mathcal{S}_1, \dots, \mathcal{S}_K), C \equiv \{\overline{W}_{S,C}^C\}$ 
3: Output: Coded messages  $V^1, \dots, V^K$ 
4:    $\mathcal{K} \leftarrow \{k : \mathcal{S}_k \notin \{\mathcal{S}_1, \dots, \mathcal{S}_{k-1}\}\}$ 
5:   for  $k \in \{1, \dots, K\}$  do
6:     for  $U \subseteq [k+1 : K] : |U| = t, \sum_{j \in \mathcal{K}} \mathbb{1}\{j \in U \cup \{k\}\} \geq 1$  do
7:        $V^k \leftarrow V^k \cup \left( \bigoplus_{j \in U \cup \{k\}} \overline{W}_{S_j, U \cup \{k\} \setminus \{j\}}^C \right)$ 
8:     end for
9:   end for
10: end function

```

---

## V. NUMERICAL RESULTS

We evaluate the performance of the scheme proposed in Subsection IV-B, referred to as the *correlation-aware* scheme, by comparing its memory-power trade-off with the lower bound presented in Theorem 1, as well as with the trade-off achieved by the scheme proposed in [12], which does not exploit the correlation among the files, referred to as the *correlation-ignorant* scheme. We consider a setting with  $N = 5$  files,  $K = 5$  users, file rate  $R = 1$ , and cache capacity  $M = 0.5$ . The channel gains are modeled as  $1/h_k^2 = 2 - 0.2(k-1)$ , for  $k = 1, \dots, 5$ . We denote by  $\alpha_\ell$  the file-length fraction that belongs to sublibrary  $L_\ell$ , i.e.,

$$\alpha_\ell = \binom{N-1}{\ell-1} \frac{R_\ell}{R}, \quad \sum_{\ell=1}^N \alpha_\ell = 1.$$

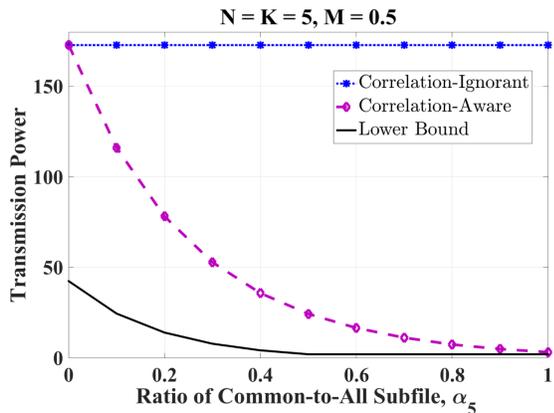


Fig. 1: Transmission power vs. common subfile fraction  $\alpha_5$ , when the files are composed of private and common-to-all subfiles.

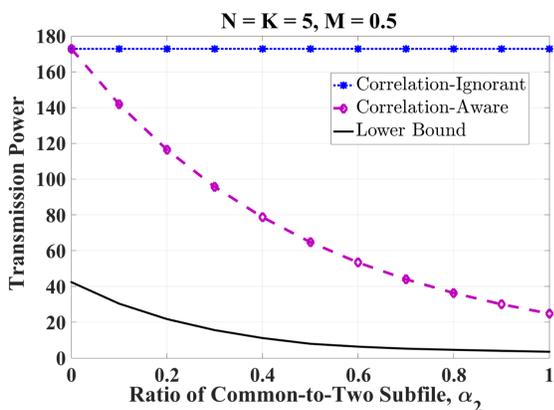


Fig. 2: Transmission power vs. common subfile fraction  $\alpha_2$ , when the files are composed of private and common-to-two subfiles.

Fig. 1 displays  $p^*(M)$ , minimum required transmit power for memory  $M$ , for a database with files composed of one *private* subfile, which is exclusive to that file, and a *common-to-all* subfile, which is shared among all files, i.e.,  $\alpha_1 + \alpha_5 = 1$ ,  $\alpha_2 = \alpha_3 = \alpha_4 = 0$ . In Fig. 2,  $p^*(M)$  is shown when the files, in addition to private subfiles, have pairwise correlations through *common-to-two* subfiles, that is  $\alpha_1 + \alpha_2 = 1$ ,  $\alpha_3 = \alpha_4 = \alpha_5 = 0$ . We plot the minimum transmit power as a function of the common parts of the files for both scenarios, i.e., with respect to  $\alpha_5$  and  $\alpha_2$ , respectively. In both settings, the transmission power achieved by the correlation-aware scheme decreases remarkably, as the portion of common subfiles increases, while the performance of the correlation-ignorant scheme does not improve. It is observed that the transmission power drops at a higher rate in Fig. 1 compared to Fig. 2 for increasing ratio of common subfiles, in both the correlation-aware scheme and lower bound. This is due to the reduction in the amount of content that is sent over the Gaussian BC, as a result of exploiting the higher level of correlation. It is also observed that the gap between the

transmit power upper and lower bounds is smaller in Fig. 1 compared to Fig. 2.

## VI. CONCLUSIONS

We have investigated caching and delivery of correlated contents over a cache-aided Gaussian BC. Correlation among files is captured by the component subfiles shared among different subsets of files. We have first presented a lower bound on the minimum transmission power which guarantees the reliable delivery of all possible demand combinations. A correlation-aware joint cache and channel coding scheme, based on superposition coding, is proposed, and the corresponding upper bound is compared numerically with the lower bound. Our numerical results show that significant energy gains can be obtained by exploiting the correlation among the files.

## REFERENCES

- [1] S. O. Somuyiwa, A. György, and D. Gündüz, "Improved policy representation and policy search for proactive content caching in wireless networks," in *Proc. IEEE Int'l Symp. on Modeling and Opt. in Mobile, Ad Hoc, and Wireless Netw. (WiOpt)*, Paris, France, May 2017.
- [2] M. A. Maddah-Ali and U. Niesen, "Fundamental limits of caching," *IEEE Trans. Inform. Theory*, vol. 60, no. 5, pp. 2856–2867, May 2014.
- [3] 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, pp. 171–175.
- [4] P. Hassanzadeh, A. Tulino, J. Llorca, and E. Erkip, "Correlation-aware distributed caching and coded delivery," in *Proc. IEEE Inform. Theory Workshop (ITW)*, Cambridge, UK, Sep. 2016, pp. 166–170.
- [5] Q. Yang and D. Gündüz, "Centralized coded caching of correlated contents," in *Proc. IEEE Int'l Conf. on Commun. (ICC)*, Kansas City, MO, May 2018.
- [6] P. Hassanzadeh, A. M. Tulino, J. Llorca, and E. Erkip, "Broadcast caching networks with two receivers and multiple correlated sources," in *Proc. IEEE Asilomar Conf. on Signals, Syst. and Computers*, Pacific Grove, CA, Oct. 2017.
- [7] —, "Rate-memory trade-off for the two-user broadcast caching network with correlated sources," in *Proc. IEEE Int'l Symp. on Inform. Theory (ISIT)*, Aachen, Germany, 2017, pp. 1227–1231.
- [8] P. Hassanzadeh, A. Tulino, J. Llorca, and E. Erkip, "Cache-aided coded multicast for correlated sources," in *Proc. IEEE Int'l Symp. on Turbo Codes and Iterative Inform. Processing (ISTC)*, Brest, France, Sep. 2016.
- [9] —, "On coding for cache-aided delivery of dynamic correlated content," to appear *IEEE Journal on Selected Areas in Communications*, 2018.
- [10] Q. Yu, M. A. Maddah-Ali, and A. S. Avestimehr, "The exact rate-memory tradeoff for caching with uncoded prefetching," *IEEE Trans. Inform. Theory*, vol. 64, no. 2, pp. 1281–1296, 2018.
- [11] M. Mohammadi Amiri and D. Gündüz, "Cache-aided content delivery over erasure broadcast channel," *IEEE Trans. Commun.*, vol. 66, no. 1, pp. 370–381, Jan. 2018.
- [12] M. Mohammadi Amiri and D. Gündüz, "Caching and coded delivery over Gaussian broadcast channels for energy efficiency," *arXiv:1712.03433v1 [cs.IT]*, Dec. 2017.
- [13] S. S. Bidokhti, M. Wigger, and R. Timo, "Noisy broadcast network with receiver caching," *arXiv:1605.02317 [cs.IT]*, May 2016.
- [14] S. S. Bidokhti, M. Wigger, and A. Yener, "Benefits of cache assignment on degraded broadcast channels," *arXiv:1605.02268v2 [cs.IT]*, Nov. 2017.
- [15] A. Ghorbel, M. Kobayashi, and S. Yang, "Content delivery over erasure broadcast channels with cache and feedback," *IEEE Trans. Inform. Theory*, vol. 62, no. 11, pp. 6407–6422, Nov. 2016.
- [16] 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, 2018.
- [17] P. P. Bergmans, "Random coding theorem for broadcast channel with degraded components," *IEEE Trans. Inform. Theory*, vol. 19, no. 2, pp. 197–207, Mar. 1973.