# Coded Caching in a Multi-Server System with Random Topology

Nitish Mital, *Student Member, IEEE*, Deniz Gündüz, *Senior Member, IEEE,* and Cong Ling, *Member, IEEE*

***Abstract*—Cache-aided content delivery is studied in a multi-server system with $P$ servers and $K$ users, each equipped with a local cache memory. In the delivery phase, each user connects randomly to any $\rho$ out of $P$ servers. Thanks to the availability of multiple servers, which model small-cell base stations (SBSs), demands can be satisfied with reduced storage capacity at each server and reduced delivery rate per server; however, this also leads to reduced multicasting opportunities compared to the single-server scenario. A joint storage and proactive caching scheme is proposed, which exploits coded storage across the servers, uncoded cache placement at the users, and coded delivery. The delivery *latency* is studied for both *successive* and *parallel* transmissions from the servers. It is shown that, with successive transmissions the achievable average delivery latency is comparable to the one achieved in the single-server scenario, while the gap between the two depends on $\rho$, the available redundancy across the servers, and can be reduced by increasing the storage capacity at the SBSs. The optimality of the proposed scheme with uncoded cache placement and MDS-coded server storage is also proved for successive transmissions.**

***Index Terms*—Coded caching, distributed storage, partial connectivity, multi-server caching, femtocaching.**

## I. Introduction

Coded caching and distributed storage have received significant attention in recent years to exploit the available memory space and processing power of individual network nodes to increase the throughput and efficiency of data availability. With proactive caching, part of the data can be pushed to nodes' local cache memories during off-peak hours, called the *placement phase*, to reduce the burden on the network during peak traffic periods, called the *delivery phase* [1] - [12]. A different type of coded caching also improves the delivery performance in the so-called "femtocaching" scenario [4], where multiple cache-equipped small-cell base stations (SBSs) collaboratively deliver contents to users. Coding for distributed storage systems has been extensively studied in the literature (see, for example, [13]), and in the femtocaching scenario, ideal maximum distance separable (MDS) codes allow users to recover contents by collecting parity bits from only a subset of SBSs they connect to [4].

In this work, we combine distributed storage at the SBSs, similar to the "femtocaching" framework [4], with cache storage at the users, and consider coded delivery over error-free shared broadcast links [2]. We consider a library of $N$ files stored across $P$ SBSs, each equipped with a limited-capacity storage space (see Fig. 1). Unlike the existing literature, we consider a boolean random connectivity model [5]: during the delivery phase, each user connects only to a random subset of $\rho$ SBSs, where $\rho \leq P$. This may be due to the density of distribution of SBSs, physical variations in the channel, or due to resource constraints. Most importantly, the connections that form the network topology are not known in advance during the placement phase; therefore, the cache placement cannot be designed for a particular network topology. Storing the files across multiple SBSs, and allowing users to connect randomly to a subset of them results in a loss in multicasting opportunities for the servers, indicating a trade-off between the coded caching gain and the flexibility provided by distributed storage across the servers, which, to the best of our knowledge, has not been studied before.

On the other hand, the presence of multiple servers may improve the latency if user requests can be satisfied in parallel. Accordingly, two scenarios are discussed depending on the delivery protocol. If the servers transmit *successively*, i.e., time-division transmission, the total latency is the sum of the latencies on each link in delivering all the requests. If the servers operate in parallel, then the latency is given by the link with the maximum latency.

We propose a practical coded storage and delivery scheme that exploits MDS coded storage across servers simultaneously with coded caching and delivery to users. In the successive transmission scenario, we show that the cost of the flexibility of distributed storage is a scaling of the latency by a constant. We also characterize the average worst-case latency (over all user-server associations) of the proposed scheme by assuming that the users connect to a uniformly random subset of the servers; and show that it is relatively close to the best-case performance, which is the single-server centralized delivery latency derived in [1], achieved when all the users connect to the same set of servers, maximizing the multicasting opportunities. We observe that, as the server storage capacities increase, the average delivery latency vs. user cache memory trade-off improves, approaching the single-server performance. We give an analytical expression to compute the average delivery latency for different server storage capacities, which is shown to give a fairly accurate estimate of the expected delivery latency when the number of servers is large. We then consider the delivery latency when the servers can transmit in parallel. We characterize the achievable average worst case delivery latency of the proposed coded storage and delivery scheme as a function of the server storage capacity for different $\rho$ values.

In a related work [10], the authors study coded caching schemes presented in [1] and [9] when parity servers are available. The authors consider special scenarios with one and two parity servers. They propose a scheme that stripes the files into blocks, and codes them across the servers with a systematic MDS code, and they also propose a scheme for the

Part of this work was presented at the IEEE Wireless Communications and Networking Conference (WCNC) in 2018 [2].

scenario in which files are stored as whole units in the servers, without striping. In our work, we do not specify servers as parity servers, and instead propose a scheme that generalizes to the use of any type of MDS code and any number of storage servers. We study the impact of the topology on the sum and maximum delivery rates, and the trade-off between the server storage capacity and the average of these rates.

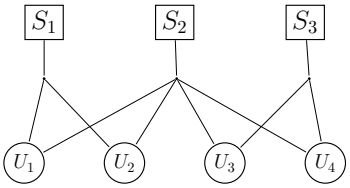
In [11], the authors consider multiple servers, each having access to all the files in the library, serving the users through an intermediate network of relays. They consider the so-called *linear network* model, in which the network topology is fixed but unknown at the relay nodes. The authors study the delivery latency considering parallel transmissions from the servers, and show that there is a gain from using multiple servers when the relay nodes employ simple random linear network coding. Note that, our model considers both limited storage servers and random network topology over the delivery network, which is unknown during the placement phase, but known during the delivery phase. Compared to the linear network model, our model corresponds to an identity network transfer matrix, in which the scheme of [11] does not provide any gains, since it is not optimized for the realization of the topology.

Another line of related works study caching in combination networks [12], [14], which consider a single server serving cache-equipped users through multiple relay nodes. The server is connected to these relays through unicast links, which in turn serve a distinct subset of a fixed number of users through unicast links. A combination network with cache-enabled relay nodes is considered in [14]. In our paper, we relax the symmetry of a standard combination network and the assumption of a fixed and known network topology, which would be unrealistic in many practical scenarios, to a certain degree by allowing each user to connect to a random fixed number of servers, thus breaking the symmetry from the servers' perspective while maintaining the symmetry from the end-users' perspective.

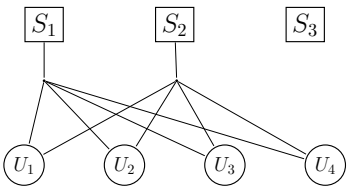
***Notations***. For two integers $i < j$, we denote the set $\{i, i+1, \ldots, j\}$ by $[i : j]$, while the set $[1 : j]$ is denoted by $[j]$. Sets are denoted with the calligraphic font, and $|\mathcal{A}|$ denotes the cardinality of set $\mathcal{A}$. For $\mathcal{A} = \{a_1, a_2, \ldots, a_p\}$, we define $X_{\mathcal{A}} \triangleq (X_{a_1}, \ldots, X_{a_p})$. $\mathbb{1}_E$ denotes the indicator function of the event $E$, i.e., its value is 1 when the event $E$ happens. $\lfloor x \rfloor$ denotes the largest integer less than or equal to $x$. $\lceil x \rceil$ denotes the smallest integer greater than or equal to $x$.

## II. Problem Setting

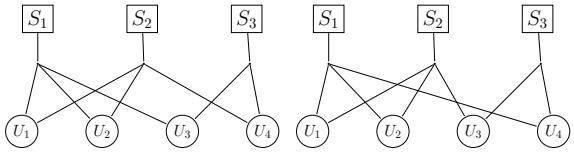
We consider the system model illustrated in Fig. 1 with $P$ servers, denoted by $\mathrm{S}_1, \mathrm{S}_2, \ldots, \mathrm{S}_P$, serving $K$ users, denoted by $\mathrm{U}_1, \mathrm{U}_2, \ldots, \mathrm{U}_K$. There is a library of $N$ files $W_1, W_2, \ldots, W_N$, each of length $F$ bits uniformly distributed over $[2^F]$. Each user has access to a local cache memory of capacity $M_U F$ bits, $0 \leq M_U \leq N$, while each server has a storage memory of capacity $M_S F$ bits. The caching scheme consists of two phases: placement phase and delivery phase. We consider a centralized placement scenario as in [1], which is carried out centrally with the knowledge of the servers and the users participating in the delivery phase. However, neither the user demands, nor the network topology is known in advance during the placement phase. In the delivery phase, we assume that each user randomly connects to $\rho$ servers out of $P$ with a uniform distribution over all $\rho-$subsets, where $\rho \leq P$, and requests a single file from the library. This is an instance of a boolean model network [5], which is a first-order approximation of isotropic wireless communication. A user connects to an SBS independently of the other SBSs, where the SBSs are assumed to be uniformly distributed, if the received SNR from that SBS is above a certain threshold, and $\rho$ is determined by the density of SBSs or other resource constraints. We define $\alpha \triangleq \frac{\rho}{P}$ as the *connectivity* of the network, where $0 \leq \alpha \leq 1$. For $j \in [K]$, let $\mathcal{Z}_j$ denote the set of servers $\mathrm{U}_j$ connects to, where $|\mathcal{Z}_j| = \rho$, and $d_j \in [N]$ denotes the index of the file it requests. For example, in Fig. 1(a), $\mathcal{Z}_1 = \{\mathrm{S}_1, \mathrm{S}_2\}$, $\mathcal{Z}_2 = \{\mathrm{S}_1, \mathrm{S}_2\}$, $\mathcal{Z}_3 = \{\mathrm{S}_2, \mathrm{S}_3\}$ and $\mathcal{Z}_4 = \{\mathrm{S}_2, \mathrm{S}_3\}$. Let the demand vector be denoted by $\mathbf{d} \triangleq (d_1, d_2, ..., d_K)$. The topology of the network, i.e., which users are connected to which servers, and the demands of the users are revealed to the servers at the beginning of the delivery phase.




(a) $\rho = 2$, $q_1 = 2, q_2 = 4, q_3 = 2$.




($b$) $\rho = 2$, $q_1 = 4, q_2 = 4, q_3 = 0$ (best topology (for successive transmissions), worst topology (for parallel transmissions)).




($c$) $\rho = 2, q_1 = 3, q_2 = 3, q_3 = 2$ (worst topology (for successive transmissions), best topology (for parallel transmissions))

Fig. 1: Examples of different network topologies for $P = 3$ and $K = 4$ with $\rho = 2$.

The complete library must be stored at the servers in a coded manner to provide redundancy, since each user connects only to a random subset of the servers. Since any user should be able to reconstruct any requested file from its own cache memory and the servers it is connected to, the total cache capacity of a user and any $\rho$ servers must be sufficient to recover the whole library; that is, we must have $M_U + \rho M_S \geq N$.

Let $\mathcal{K}_p$ denote the set of users served by $\mathrm{S}_p$, for $p \in [P]$,

and define the random variable $Q_p \triangleq |\mathcal{K}_p|$, which denotes the number of users served by $\mathrm{S}_p$. We shall denote a particular realization of $Q_p$ as $q_p$ and define $\mathbf{q} \triangleq (q_1, \ldots, q_p)$, where we have $\sum_{p=1}^{P} q_p = K\rho$. For example, in Fig. 1(a), we have $\mathcal{K}_1 = \{\mathrm{U}_1, \mathrm{U}_2\}, \mathcal{K}_2 = \{\mathrm{U}_1, \mathrm{U}_2, \mathrm{U}_3, \mathrm{U}_4\}, \mathcal{K}_3 = \{\mathrm{U}_3, \mathrm{U}_4\}$, and $\mathbf{q} = (2, 4, 2)$. In the delivery phase, server $\mathrm{S}_p$ transmits message $X_p$ of size $R_pF$ bits to the users connected to it, i.e., the users in set $\mathcal{K}_p$, over the corresponding shared link. We assume that each server is allocated a separate orthogonal delivery channel, and the message it transmits is received by all the users connected to this server. The message $X_p$ is a function of the demand vector $\mathbf{d}$, the network topology, the storage contents of server $\mathrm{S}_p$, and the cache contents of the users in $\mathcal{K}_p$. User $U_k$ receives the messages $X_{\mathcal{Z}_k} \triangleq \{X_p : p \in \mathcal{Z}_k\}$, and reconstructs its requested file $W_{d_k}$ using these messages and its local cache contents.

### A. Formal Problem Statement

We now provide the formal definition of the caching problem. Let $\{W_n\}_{n=1}^{N}$ be $N$ independent random variables each uniformly distributed over $[2^F]$ for some $F \in \mathbb{N}$. Each $W_n$ represents a file of size $F$ bits. Let $R_p, p \in [P]$, be the number of bits, normalized by the size of a file, transmitted by server $p \in [P]$ during the delivery phase. A $(M_S, M_U, R_1, \ldots, R_P)$ storage and caching scheme consists of $P$ server storage functions, $K$ caching functions, $P\binom{P}{\rho}^K$ encoding functions, and $K\binom{P}{\rho}^K$ decoding functions.

The caching function

$$\phi_k : [2^F]^N \to [2^{\lfloor FM_U \rfloor}], \quad k \in [K], \tag{1}$$

maps the library $\{W_n\}_{n=1}^{N}$ into the cache contents, of user $U_k$ during the placement phase:

$$V_k \triangleq \phi_k(W_1, \ldots, W_N) \tag{2}$$

The server storage function

$$\sigma_p : [2^F]^N \to [2^{\lfloor FM_S \rfloor}], \quad p \in [P] \tag{3}$$

maps the library $\{W_n\}_{n=1}^{N}$ into the storage of server $S_p$:

$$Y_p \triangleq \sigma_p(W_1, \ldots, W_N). \tag{4}$$

We define a separate encoding function for each server depending on the network topology. Hence, the encoding function for server $S_p, p \in [P]$,

$$\psi^p_{\{\mathcal{K}_p\}_{p=1}^{P}} : [N]^K \times [2^{\lfloor FM_S \rfloor}] \to [2^{\lfloor FR_p \rfloor}] \tag{5}$$

maps the demand vector and the memory contents of server $S_p$ to message $X_p$, i.e.,

$$X_p \triangleq \psi^p_{\{\mathcal{K}_p\}_{p=1}^{P}}(\mathbf{d}, Y_p), \tag{6}$$

which is delivered to the users in $\mathcal{K}_p$ during the delivery phase. Finally, we define a separate decoding function for each user depending on the network topology. Hence, the decoding function for user $U_k, k \in [K]$, is

$$\mu^k_{(\{\mathcal{Z}_k\}_{k=1}^{K})} : [N]^K \times [2^{\lfloor FR_{\pi^k(1)} \rfloor}] \times \cdots \times [2^{\lfloor FR_{\pi^k(\rho)} \rfloor}] \times [2^{\lfloor FM_U \rfloor}] \to [2^{\lfloor F \rfloor}], \tag{7}$$

where $\pi^k(1), \ldots, \pi^k(\rho)$ denote the $\rho$ servers in set $\mathcal{Z}_k$, maps the demand vector $\mathbf{d}$, the received signals $X_{\mathcal{Z}_k}$ from the servers in $\mathcal{Z}_k$, and the local cache content $V_k$ to the estimate $\hat{W}_{d_k}$, i.e.,

$$\hat{W}_{d_k} \triangleq \mu^k_{(\{\mathcal{Z}_k\}_{k=1}^{K})}(\mathbf{d}, X_{\mathcal{Z}_k}, V_k) \tag{8}$$

The probability of error for this scheme, for a fixed topology, is defined as

$$\max_{\mathbf{d} \in [N]^K} \max_{k \in [K]} \Pr(\hat{W}_{d_k} \neq W_{d_k}). \tag{9}$$

We remark here that the storage and caching functions $\sigma_p$ and $\phi_k$ do not depend on the network topology, while the encoding and decoding functions do.

**Definition 1.** *The tuple $(M_S, M_U, R_1, \ldots, R_P)$ is said to be achievable if for every $\epsilon > 0$ and large enough file size $F$ there exists a $(M_S, M_U, R_1, \ldots, R_P)$ caching scheme with probability of error less than $\epsilon$.*

Our goal is to minimize the delivery latency, which is the time by which all the user requests can be satisfied. Among other parameters, delivery latency also depends on the operation of the SBSs. If each SBS transmits over an orthogonal frequency band, the requests can be delivered in parallel, and the delivery latency is given by $T_{pd} = \max_p R_p$. If, instead, the servers transmit successively in a time-division manner, which is suitable for user devices that are simple and not capable of multihoming on multiple frequencies, the normalized delivery latency will be given by $T_{sd} = \sum_{p=1}^{P} R_p$. Our goal will be to find the average worst-case delivery latency, where the worst case refers to the fact that all the users can correctly decode their requested files, independent of the combination of files requested by them, and the averaging is over all possible network topologies. Assuming that $N \geq K$ (i.e., the number of files is larger than the number of users), it is not difficult to see that all the users requesting a different file corresponds to the worst-case scenario. We would also like to remark that, under uniform file popularity, the probability of experiencing this worst-case demand distribution increases significantly with $N$, and approaches 1 for $N$ values that one expects to experience in practice.

## III. Coded Distributed Storage and Caching Scheme

We first note that our system model brings together aspects of distributed storage and proactive caching/coded delivery. To see this, consider the system without any user caches, i.e., $M_U = 0$, which is equivalent to a distributed storage system with unreliable servers, where random $P - \rho$ out of $P$ servers are inactive. It is known that MDS codes provide much higher reliability and efficiency compared to replication in this scenario [13]. On the other hand, when the servers are reliable, i.e., $\rho = P$, our system is equivalent to the one in [1], and coded delivery provides significant reductions in the delivery latency. Accordingly, our proposed scheme brings together benefits from coded storage and coded delivery. To illustrate the main ingredients of the proposed scheme we assume $M_S = \frac{N}{\rho}$ in this section, and extend to other server capacities in later sections.

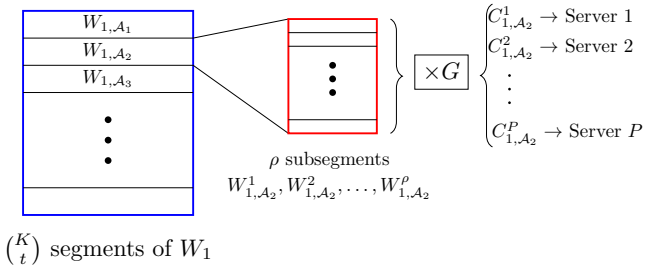



Fig. 2: Segmentation, MDS coding and placement of files.

### A. Server Storage Placement

We first describe how the files are stored across the SBS servers in order to guarantee that each user request can be satisfied from any $\rho$ servers a user may connect to (see Fig. 2). We define $t \triangleq \frac{KM_U}{N}$, and assume initially that $t$ is an integer, i.e., $t \in [0 : M_U]$. The solution for non-integer $t$ values will be obtained through memory-sharing [1]. Each file is divided into $\binom{K}{t}$ equal-size non-overlapping segments. We enumerate them according to distinct $t$-element subsets of $[K]$, where $W_{j,\mathcal{A}}$ denotes the segment of $W_j$ that corresponds to subset $\mathcal{A}$. We have $W_j = \bigcup_{\mathcal{A}\subset[K]:|\mathcal{A}|=t} W_{j,\mathcal{A}}$, $j \in [N]$.

Each segment is further divided into $\rho$ equal-size non-overlapping sub-segments denoted by $W^l_{j,\mathcal{A}}$, $l \in [\rho]$. The $\rho$ sub-segments of each segment are coded together using a $(P, \rho)$ linear MDS code with generator matrix $G$, giving as output $P$ coded subsegments for segment $W_{j,\mathcal{A}}$, denoted by $C^l_{j,\mathcal{A}}, l \in [P]$. $C^l_{j,\mathcal{A}}$ is a linear combination of the sub-segments of the segment corresponding to subset $\mathcal{A}$, of file $W_j$. $C^l_{j,\mathcal{A}}$ will be stored in server $\mathrm{S}_l$, $\forall l \in [P], j \in [N]$, and $\mathcal{A} \subset [K], |\mathcal{A}| = t$. Since each sub-segment is of length $\frac{F}{\rho\binom{K}{t}}$, every linear combination $C^l_{j,\mathcal{A}}$ is of the same length; and hence, server storage capacity constraint of $M_S F = \frac{NF}{\rho}$ is met with equality.

**Remark 2.** *We assume that each user knows the generator matrix of the MDS code to be able to reconstruct any coded subsegment $C^l_{j,\mathcal{A}}$ from the uncoded segment $W_{j,\mathcal{A}}$.*

### B. User Cache Placement

Using the placement scheme proposed in [1] for user caches, each segment of a file, $W_{j,\mathcal{A}}$, is placed into the caches of all the users $\mathrm{U}_k$ for which $k \in \mathcal{A}$, i.e., each user caches $\binom{K-1}{t-1}$ segments of each file, or $\frac{\binom{K-1}{t-1}}{\binom{K}{t}} NF = \frac{t}{K} N = M_U F$ bits, meeting the user cache capacity constraint.

### C. Delivery Phase

We first make the following observation about the above placement scheme: in the worst-case demand scenario, consider any $t+1$ users. Any $t$ out of these $t+1$ users share in their caches one segment of the file requested by the remaining user. Enumerate these subsets of $t+1$ users as $\mathcal{H}_i$, $i \in \left[\binom{K}{t+1}\right]$. Consider server $\mathrm{S}_p$, $p \in [P]$, and one of the $q_p$ users connected to it, say $U_k$. Then, for any subset $\mathcal{H}_i$, that includes $k$, i.e., $k \in \mathcal{H}_i$, the segment $W_{d_k,\mathcal{H}_i\setminus\{k\}}$ is needed by user $\mathrm{U}_k$, but is not available in its cache because $k \notin \mathcal{H}_i \setminus \{k\}$, while it is available in the caches of the users in $\mathcal{K}_p \bigcap \mathcal{H}_i \setminus \{k\}$. The MDS coded subsegment of $W_{d_k,\mathcal{H}_i\setminus\{k\}}$ stored by $\mathrm{S}_p$ is $C^p_{d_k,\mathcal{H}_i\setminus\{k\}}$, and since the users know the generator matrix $G$, each user which has $W_{d_k,\mathcal{H}_i\setminus\{k\}}$ in its cache can reconstruct $C^p_{d_k,\mathcal{H}_i\setminus\{k\}}$ as well. Then, for each $\mathcal{H}_i$ that includes at least one user from $\mathcal{K}_p$, $\mathrm{S}_p$ transmits

$$X_p(\mathcal{H}_i) = \bigoplus_{k\in\mathcal{K}_p \bigcap \mathcal{H}_i\setminus\{k\}} C^p_{d_k,\mathcal{H}_i\setminus\{k\}}, \tag{10}$$

where $\bigoplus$ denotes the bitwise XOR operation. Then, $\left|\left\{i \in \left[\binom{K}{t+1}\right] : k \in \mathcal{H}_i\right\}\right| = \binom{K-1}{t}$ is the number of messages transmitted by server $\mathrm{S}_p$ that contain the coded version of a segment requested by $\mathrm{U}_k$, and is also equal to the number of segments of $W_{d_k}$ not present in the cache of user $\mathrm{U}_k$. Overall, the message transmitted by $S_p$ is given by

$$X_p = \bigcup_{i\in\left[\binom{K}{t+1}\right]:\mathcal{K}_p \bigcap \mathcal{H}_i \neq \phi} X_p(\mathcal{H}_i). \tag{11}$$

From the transmitted message $X_p(\mathcal{H}_i)$ in (10) for each set $\mathcal{H}_i$, user $\mathrm{U}_k$ can decode the MDS coded version $C^p_{d_k,\mathcal{H}_i\setminus\{k\}}$ of its requested segment $W_{d_k,\mathcal{H}_i\setminus\{k\}}$. With the transmissions from all the servers, $\mathrm{U}_k$ receives $\rho$ coded versions of each missing segment from the $\rho$ servers it is connected to. Since each segment is coded with a $(P, \rho)$ MDS code, the user is able to decode each missing segment of its request.

Note that each transmitted message $X_p(\mathcal{H}_i)$ by a server is of length $F/\rho\binom{K}{t}$ bits. The number of messages transmitted by $\mathrm{S}_p$ is

$$\left|\left\{i \in \left[\binom{K}{t+1}\right] : \mathcal{K}_p \bigcap \mathcal{H}_i \neq \phi\right\}\right| \tag{12}$$

$$= \binom{K}{t+1} - \left|\left\{i \in \left[\binom{K}{t+1}\right] : \mathcal{K}_p \bigcap \mathcal{H}_i = \phi\right\}\right| \tag{13}$$

$$= \binom{K}{t+1} - \binom{K-q_p}{t+1}. \tag{14}$$

That is, server $\mathrm{S}_p$ transmits a total of $R_p = {}^{F}/_{\rho\binom{K}{t}} \left[\binom{K}{t+1} - \binom{K-q_p}{t+1}\right]$ bits.

The delivery latency performance of this proposed coded storage and delivery scheme with both successive and parallel SBS transmissions will be studied in the following two sections.

**Remark 3.** *Due to the symmetry in the network across servers and users, the delivery latency of this scheme depends only on the* $\mathbf{q}$ *vector, not the particular network topology, i.e., what matters is the number of users served by each server, not the identity of the users. More specifically, all permutations of a* $\mathbf{q}$ *vector, and the associated users, result in the same latency. Hence, we define the "type" of a network topology as a vector of dimension $K+1$,* $\mathbf{g}$*, where $g_i$ denotes the number of servers serving $i$ users, for $i = 0, 1, \ldots, K$. We have $0 \leq g_i \leq P$, $\sum_{i=0}^{K} g_i = P$ and $\sum_{i=0}^{K} i g_i = K\rho$.*

## IV. Successive SBS Transmissions

In this section we assume that the SBSs share the same communication resources, and hence, transmit successively to avoid interference. When the SBSs transmit successively in time, the normalized delivery latency is given by

$$T_{sd} \triangleq \sum_{p=1}^{P} R_p = \frac{1}{\rho\binom{K}{t}} \sum_{p=1}^{P} \left[ \binom{K}{t+1} - \binom{K-q_p}{t+1} \right] \quad (15)$$

$$= \frac{1}{\alpha} \frac{(K-t)}{(t+1)} - \frac{1}{\rho\binom{K}{t}} \sum_{p=1}^{P} \binom{K-q_p}{t+1} \quad (16)$$

$$= \frac{1}{\alpha} \frac{(K-t)}{(t+1)} - \frac{1}{\rho\binom{K}{t}} \sum_{i=0}^{K} g_i \binom{K-i}{t+1}. \quad (17)$$

To characterize the "best" and "worst" network topologies that lead to the minimum and maximum delivery latency, respectively, we present the following lemma without proof.

**Lemma 4.** *For $n_1, n_2, r \in \mathcal{Z}^+$ satisfying $r \leq n_1$ and $n_1+2 \leq n_2$, we have*

$$\binom{n_1}{r} + \binom{n_2}{r} \geq \binom{n_1+1}{r} + \binom{n_2-1}{r}. \quad (18)$$

The lemma above indicates the "convex" nature of the binomial coefficients in (16); that is, the points $(r, \binom{r}{r})$, $(r+1, \binom{r+1}{r})$, …, $(n_1+n_2-r, \binom{n_1+n_2-r}{r})$ form a convex region. From Lemma 4, it can be deduced that the second summation term in (16) takes its minimum when $\max_p(q_p) \leq \min_p(q_p)+1$, $p \in [P]$, i.e., the values of $q_p$ are as close to each other as possible. This corresponds to the class of topologies with the highest delivery latency (see Fig. 1(c) for an example). The topology that requires the minimum delivery latency of $T_{sd} = \frac{K-t}{t+1}$ is when $q_p$ is either 0 or $K$ for each server, or equivalently, when all the users are connected to the same $\rho$ servers (see Fig. 1(b) for an example).

Next we study the average worst-case normalized delivery latency, where the average is taken over all possible network topologies. As we have seen above, the delivery latency depends on the topology, and for a given topology, the "worst-case" delivery latency refers to the worst-case demand combination when each user requests a different file. Note that, in the worst case, due to the symmetry in the network and the proposed caching and delivery scheme, the latency depends only on the type of the network topology. We further assume that the probability of having any network of the same type is the same.

**Lemma 5.** *Let $w_i$ be the probability of exactly $i$ users being served by a server; that is, $w_i = Pr\{q_p = i\}, p \in [P]$. We have*

$$\mathbb{E}[g_i] = w_i P. \quad (19)$$

*Proof:* The number of servers serving exactly $i$ users, $g_i$, can be written as

$$g_i = \sum_{p=1}^{P} \mathbb{1}_{\{q_p = i\}}. \quad (20)$$

Taking expectation on both sides, we have

$$\mathbb{E}[g_i] = \sum_{p=1}^{P} \Pr\{q_p = i\} \quad (21)$$

$$= w_i P. \quad (22)$$

■

The following theorem presents the average normalized worst-case delivery latency of the proposed scheme under successive transmissions, which follows by taking the expectation of both sides of Eq. (17) and Lemma 5.

**Theorem 6.** *The average worst-case normalized delivery latency of the proposed scheme over all topologies under random user-server association is given by*

$$\mathbb{E}[T_{sd}] = \frac{1}{\alpha} \frac{(K-t)}{(t+1)} - \frac{1}{\alpha\binom{K}{t}} \sum_{i=0}^{K} w_i \binom{K-i}{t+1}. \quad (23)$$

Since we have assumed uniform random connectivity, we have $w_i = \binom{K}{i}\binom{P-1}{\rho-1}^i \binom{P-1}{\rho}^{K-i} / \binom{P}{\rho}^K = \binom{K}{i}\alpha^i(1-\alpha)^{K-i}$. The average worst-case latency is given in the following corollary.

**Corollary 7.** *The average worst-case normalized delivery latency with successive transmissions under uniformly random user-server association is given by*

$$\mathbb{E}[T_{sd}] = \frac{K-t}{t+1} \left[ \frac{1-(1-\alpha)^{t+1}}{\alpha} \right]. \quad (24)$$

*Proof: By plugging in $w_i = \binom{K}{i}\alpha^i(1-\alpha)^{K-i}$ in Eq. (23), we obtain the above simplified expression.* ■

### A. Redundancy in Server Storage Capacity

In the analysis above, we have set the server storage capacity to $M_S = \frac{N}{\rho}$. On the other hand, for a given user cache capacity $M_U$, the minimum server storage capacity that would allow the reconstruction of any demand combination is given by $M_S = \frac{N-M_U}{\rho}$. In this case, we cache the same $\frac{M_U}{N}$ fraction of the library in all the user caches during the placement phase, and deliver the remaining fraction of the demands from the servers, which is identical to the scheme in [14] when the user and its connected servers have just enough space to store the entire library. The worst-case delivery latency in this case is given by $T_{sd} = K\left(1 - \frac{M_U}{N}\right) = K - t$.

Next, we consider the case when there is redundancy in server memories; that is, $\frac{N}{\rho} < M_S \leq N$. Assume that $M_S = \frac{N}{\rho-z}$ for some integer $z \in [\rho - 1]$. Define $\hat{\alpha} \triangleq \frac{\rho-z}{P}$. Since $\alpha$ is defined as the connectivity of the network, $\alpha - \hat{\alpha}$ is the storage redundancy. For non-integer values of $z$, the solution can be obtained by memory-sharing.

In this case, a $(P, \rho-z)$ MDS code is used for server storage placement, allowing each user to reconstruct any requested file by connecting to $\rho - z$ servers. The user cache placement is done as in the previous section. In the delivery phase, each user randomly connects to $\rho$ servers. We now have a degree of freedom thanks to the additional storage space available at each server. Each user can obtain a segment from any $\rho - z$ of the $\rho$ servers it is connected to by receiving one coded subsegment from each of them. The choice of the

$$\begin{array}{c} \\ S_1 \\ S_2 \\ S_3 \\ S_4 \\ S_5 \\ S_6 \\ S_7 \end{array} \begin{array}{c} \begin{array}{ccccc} U_1 & U_2 & U_3 & U_4 & U_5 \end{array} \\ \left[\begin{array}{ccccc} 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 \end{array}\right] \end{array}$$

Fig. 3: An example $7 \times 5$ incidence matrix ($P = 7, K = 5$) with $\rho = 4$.

servers that deliver the coded subsegments to the users is made such that the multicasting opportunities across the network are maximized. We construct an incidence matrix $A$ of dimensions $P \times K$ such that $a_{ij} = 1$ if $S_i$ is connected to $U_j$, $a_{ij} = 0$ otherwise. Consider the $(t+1)-$element subset $\mathcal{H}_i$, and the file segments $W_{d_k,\mathcal{H}_i\backslash\{k\}}, \forall k \in \mathcal{H}_i$. Consider the columns of $A$ corresponding to the users in $\mathcal{H}_i$ and the matrix $Q$ formed by them. Define the minimum cover of $\mathcal{H}_i$ as the smallest $l$ for which a $l \times (t+1)$ submatrix of $Q$ has at least $\rho - z$ non-zero values in each column. The servers corresponding to the $l$ rows of this submatrix have to transmit one coded message each to satisfy the requests for the missing segments corresponding to $\mathcal{H}_i$. Therefore, the total number of transmissions required to deliver the segments $W_{d_k,\mathcal{H}_i\backslash\{k\}}, k \in \mathcal{H}_i$, is equal to the minimum cover of $\mathcal{H}_i$.

As an example, consider the incidence matrix as shown in Fig. 3, which corresponds to a system with $P = 7$ servers and $K = 5$ users, where each user connects to $\rho = 4$ servers. Assume that the server storage capacity is $M_S = \frac{N}{\rho-2}$ and $t = 1$. In this setting, coded subsegments of requested files can be delivered to $t+1 = 2$ users through multicasting, and it is sufficient for each user to receive coded segments from $\rho - 2 = 2$ servers. Then, for the user set $\mathcal{H}_i = \{1, 2\}$, we consider the submatrix corresponding to the columns 1 and 2 and rows 1 and 2 (marked by the blue dashed lines in Fig. 3), which is the smallest submatrix satisfying the condition that each column has at least $\rho - z = 2$ 1s. Hence, the minimum cover for $\mathcal{H}_i = \{1, 2\}$ is equal to the number of rows of this submatrix, that is, 2. For $\mathcal{H}_i = \{3, 4\}$ (marked by the red dashed lines in Fig. 3), the minimum cover is 3. Thus, from (10), for segments $W_{d_k,\{3,4\}\backslash\{k\}}, k \in \{3, 4\}$, $S_3$ transmits the message $X_3(\{3, 4\}) = \bigoplus_{k\in\{3,4\}} C^3_{d_k,\{3,4\}\backslash\{k\}}$, $S_4$ transmits $X_4(\{3, 4\}) = C^4_{d_3,\{4\}}$, and $S_5$ transmits $X_5(\{3, 4\}) = C^5_{d_4,\{3\}}$. The total number of transmissions is 3. We can go through all the $(t+1)-$ element subsets of the users and identify for each of them the minimum cover. We note that in the successive transmission scenario, the total latency does not depend on the server transmitting each subsegment, since the contribution to the total latency is the same. In the above example servers $S_1$ and $S_6$ could also deliver the two coded subsegments to users $U_1$ and $U_2$. The selection of the servers matters in the case of parallel transmissions.

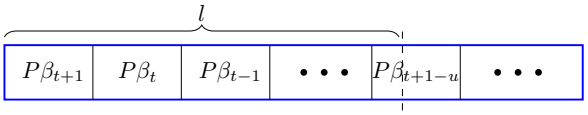



Fig. 4: The ordering of servers to count the minimum cover. The dashed line indicates the point at which enough servers have been counted to deliver $\hat{\alpha}$ coded subsegments to all users in $\mathcal{H}$.

### B. Performance analysis

In this section, we derive an analytical expression for the expected delivery latency in the asymptotic regime, i.e., when $P \to \infty$, while $\alpha$ and $\hat{\alpha}$ are fixed. Consider a particular subset $\mathcal{H}$ of $t+1$ users. Define $\beta_i$ as the fraction of servers serving $i$ users in $\mathcal{H}$, $i = 0, 1, \ldots, t+1$. Thus, we have

$$\beta_i = \frac{1}{P}\sum_{p=1}^{P} \mathbb{1}_{\{|\mathcal{H}\cap\mathcal{K}_p|=i\}}. \tag{25}$$

Taking expectation on both sides of Eq. (25), we have

$$\mathbb{E}[\beta_i] = \frac{1}{P}\sum_{p=1}^{P} \mathbb{E}[\mathbb{1}_{\{|\mathcal{H}\cap\mathcal{K}_p|=i\}}] \tag{26}$$

$$= \frac{1}{P}\sum_{p=1}^{P} \Pr(|\mathcal{H}\cap\mathcal{K}_p| = i) \tag{27}$$

$$= \Pr(|\mathcal{H}\cap\mathcal{K}_p| = i) \tag{28}$$

$$= \binom{t+1}{i}\alpha^i(1-\alpha)^{(t+1-i)}, \tag{29}$$

where (28) follows due to the symmetry across all the servers. By the law of large numbers, $\beta_i \to \mathbb{E}[\beta_i]$ for all $i \in [K]$, as $P \to \infty$. Also, the topology becomes symmetric across all users as $P \to \infty$, i.e., almost all user subsets of the same size are served by the same number of servers. We group the servers serving the same number of users and arrange them in the order as illustrated in Fig. 4, where the first $P\beta_{t+1}$ servers serve $t+1$ users in $\mathcal{H}$, the next $P\beta_t$ servers serve exactly $t$ users in $\mathcal{H}$, and so on. To compute the minimum cover $l$, i.e., the minimum number of servers that are needed to deliver $\hat{\alpha}$ coded subsegments to each user in $\mathcal{H}$, we start counting from the left, until each user in $\mathcal{H}$ collects $\hat{\alpha}$ coded subsegments. For some $u \in [0 : t]$, we count till the $(u+1)$-th set of servers which serve $t+1-u$ users in $\mathcal{H}$, as in Fig. 4. When counting the set of servers serving $t+1-u$ users, note that, according to our scheme, the $t+1-u$ users can each extract one coded subsegment from a message transmitted by a server in that set. Therefore, $\lceil \frac{t+1}{t+1-u} \rceil$ servers are required to serve one coded subsegment each to the $t+1$ users in $\mathcal{H}$. Define $\delta$ as the number of coded subsegments required by a single user in $\mathcal{H}$ from the set of servers serving $t+1-u$ users. Therefore, for $P \to \infty$, the minimum cover can be written as

$$l \approx P\sum_{j=0}^{u-1} \beta_{t+1-j} + \delta\left\lceil \frac{t+1}{t+1-u} \right\rceil \tag{30}$$

for some $u \in [0 : t]$, where (30) follows thanks to the symmetry across users. Note that the above analysis is asymptotic,

and does not hold in general for a finite $P$. Since a message transmitted by a server serving $i$ users in $\mathcal{H}$ delivers $i$ coded subsegments in total to the $i$ users, the total number of coded subsegments delivered by the $l$ servers that form the minimum cover for the users in $\mathcal{H}$ must be at least $(t+1)\hat{\alpha}$; that is,

$$P\sum_{j=0}^{u-1}(t+1-j)\beta_{t+1-j}+\delta^{'}\left\lceil\frac{t+1}{t+1-u}\right\rceil\geq(t+1)P\hat{\alpha}, \tag{31}$$

where $\delta^{'}\triangleq(t+1-u)\delta$. The value of $u$ is determined by solving for

$$\begin{aligned}0\leq(t+1)P\hat{\alpha}-P\sum_{j=0}^{u-1}(t+1-j)\beta_{t+1-j}\\ \leq(t+1-u)\beta_{t+1-u}.\end{aligned} \tag{32}$$

From Eq. (29) and the asymptotic convergence of $\beta_i$ to its expectation, we have

$$\begin{aligned}&\sum_{j=0}^{u-1}(t+1-j)\beta_{t+1-j}\\ &\overset{P\to\infty}{\to}\sum_{j=0}^{u-1}\binom{t+1}{t+1-j}(t+1-j)\alpha^{(t+1-j)}(1-\alpha)^{j}\\ &=\alpha^{t+1}(t+1)\sum_{j=0}^{u-1}\binom{t}{s}\left(\frac{\alpha}{1-\alpha}\right)^{-j}\end{aligned} \tag{33}$$

We substitute (33) into (32) to solve for $u$. Having first determined $u$ from Eq. (32), and then $\delta$ from (31), we can find the minimum cover $l$ from Eq. (30) for $P\to\infty$. The delivery latency can thus be estimated as

$$\mathbb{E}[T_{sd}]=\frac{1}{(\rho-z)}\left(\frac{K-t}{t+1}\right)l, \tag{34}$$

where the factor $\frac{1}{(\rho-z)}\left(\frac{K-t}{t+1}\right)$ is obtained by multiplying the normalized size of each coded subsegment, given by $\frac{1}{(\rho-z)\binom{K}{t}}$, with the number of $(t+1)-$user subsets, given by $\binom{K}{t+1}$. It will be seen in Section VII that Eq. (34) provides a fairly accurate estimate of the expected delivery latency when the number of servers $P$ is large.

## V. Lower Bound

In this section, we derive a tight lower bound on the minimum expected delivery latency with uncoded cache placement, coded distributed storage in the servers, and successive transmissions, which shows the optimality of the caching and delivery scheme proposed in Section IV in certain regimes. Following [15], we will first represent the problem as a set of index coding problems.

In the index coding problem [17], a sender wishes to communicate an independent message $M_j, j\in[B]$, uniformly distributed over $[2^{nr_j}]$, to the $j^{th}$ user among $B$ users by broadcasting a message $X^n$ of length $n$. Each user $j$ knows a subset of the messages targeting these $B$ users, indicated by $\mathcal{B}_j, \mathcal{B}_j\subset\{M_1,\ldots,M_B\}$, referred to as side information. A rate tuple $(r_1,\ldots,r_B)$ is achievable, for large enough $n$, if every user can restore its desired message with high probability based on $X^n$ and its side information. The index coding problem can be represented as a directed graph $G$ with $B$ nodes, where node $i$ represents message $M_i$, and a directed edge connects node $i$ to node $j$ if user $j$ knows message $M_i$ as side information. For our problem setting, where we have the file library $\{W_i\}_{i=1}^{N}$, each file $W_i, i\in[N]$, of size $F$ bits is divided into $2^K$ non-overlapping segments denoted by $W_{i,\mathcal{A}}$, $\mathcal{A}\in 2^{[K]}$, where $2^{[K]}$ indicates the power set $\{\phi,\{1\},\{2\},\{3\},\{1,2\}\ ,\ldots,[K]\}$. The segment $W_{i,\mathcal{A}}$ denotes the part of file $W_i$ cached exclusively by users in set $\mathcal{A}$. This is the most general representation of an uncoded caching scheme at the users. For each demand vector $\mathbf{d}$ with distinct requests, corresponding to the worst case scenario, we consider an index coding problem with $K2^{K-1}$ independent messages, each of which represents a segment requested by a particular user and cached by a different subset of the remaining users.

We generate a directed graph with $K2^{K-1}$ nodes corresponding to these messages, such that, for $i\neq j$ and $\mathcal{A}_i\subset[K]\setminus\{i\}$ and $\mathcal{A}_j\subset[K]\setminus\{j\}$, there is a directed edge from node $W_{d_i,\mathcal{A}_i}$ to $W_{d_j,\mathcal{A}_j}$, $i\neq j$, if user $U_j$ caches the segment $W_{d_i,\mathcal{A}_i}$; that is, if $j\in\mathcal{A}_i$. In the single server centralized setting, we get a lower bound using the index coding bound [16]. Multi-server index coding has been studied as the distributed index coding problem in [18], [19]. In the distributed index coding problem, the servers are considered to store a subset of the messages in uncoded form, and each user is connected to all the servers, whereas in our problem each user can connect to $\rho$ out of $P$ servers randomly. Therefore, for the user to be able to retrieve any requested file from the servers it connects to, the files must be stored using a distributed storage scheme in the servers. Therefore, we analyse the case where the files are stored using erasure codes in the servers. In that, we encode the segment $W_{d_i,\mathcal{A}}$ into $P$ distinct coded subsegments, denoted by $(C^1_{d_i,\mathcal{A}},\ldots,C^P_{d_i,\mathcal{A}})\in\prod_{p\in[P]}\left[2^{n_p r^p_{d_i,\mathcal{A}}}\right]$, where $n_p=FR_p$ is the length of message in bits transmitted by server $S_p$, such that any $\rho$ coded subsegments can be used to reconstruct the original segment. $r^p_{d_i,\mathcal{A}}$ is the rate at which server $S_p$ transmits the coded subsegment $C^p_{d_i,\mathcal{A}}$ corresponding to user $U_i's$ request, and we have $\sum_{j=1}^{\rho}n_{\pi(j)}r^{\pi(j)}_{d_i,\mathcal{A}}\geq|W_{d_i,\mathcal{A}}|$ as a necessary condition to ensure that the segment $|W_{d_i,\mathcal{A}}|$ can be reconstructed by receiving any $\rho$ distinct coded subsegments, where $\pi(j), j\in[\rho]$, are the $\rho$ servers in set $\mathcal{Z}_i$. Recall that $\mathcal{Z}_i$ is the set of $\rho$ servers that serve user $U_i$. Also note that we do not code across files, but encode each file separately.

For the multi-server scenario, we consider $P$ index coding problems, each represented as a distinct directed graph $G_p, p\in[P]$. Each node in $G_p$ corresponds to a distinct coded subsegment $C^p_{d_i,\mathcal{A}}$, which is requested by user $U_i$ and available in server $S_p$. By distinct coded subsegments we mean that $H(W_{d_i,\mathcal{A}}|C^p_{d_i,\mathcal{A}},C^q_{d_i,\mathcal{A}})<H(W_{d_i,\mathcal{A}}|C^p_{d_i,\mathcal{A}})$ for all $p,q\in[P], p\neq q$. $G_p$ has the same structure as $G$, with the subsegments requested by users not served by server $S_p$ removed. Let the set of nodes in the index coding problem

represented by graph $G_p$ be denoted by $\mathcal{I}_p$.

We have the following multi-server index coding bound applying the result in [16] separately on each of the $P$ index coding problems.

**Theorem 8.** *If the rate tuple* $\{r^1_{1,\mathcal{A}},\ldots,r^1_{K,\mathcal{A}},\ldots,r^p_{1,\mathcal{A}},\ldots,r^p_{K,\mathcal{A}},\ldots,r^P_{1,\mathcal{A}},\ldots,r^P_{K,\mathcal{A}}\}_{\mathcal{A}\subseteq[K]}$ *is achievable for the multi-server index coding problem represented by the set of directed graphs* $G_p, p=1,\ldots,P$, *under the constraint* $\sum_{j=1}^{\rho} n_{\pi(j)} r^{\pi(j)}_{d_i,\mathcal{A}} \geq |W_{d_i,\mathcal{A}}|$, *and inter-file coding is not allowed, then* $r^p_{j,\mathcal{A}}=0$ *if server* $S_p$ *does not serve user* $U_j$, *and*

$$\sum_{p=1}^{P}\sum_{\mathcal{J}_p} r^p_{j,\mathcal{A}} \leq 1 \tag{35}$$

*for all* $\mathcal{J}_p \subseteq \mathcal{I}_p$ *where the subgraph of* $G_p$ *over* $\mathcal{J}_p$ *does not contain a directed cycle.*

**Remark 9.** *Theorem 8 holds when the nodes in the P index coding problems correspond to distinct coded subsegments, that is, there are no repeating nodes in any two index coding problems. Distributed storage schemes which concatenate repetition codes with other storage codes may not satisfy the bound in Theorem 8 (for example, see [22]). References [18] and [19] may indicate how to compute the capacity under such distributed storage schemes, but they are outside the scope of this paper.*

**Remark 10.** *There are non-MDS distributed storage codes, called regenerating codes, that utilize increased storage capacity on the servers to reduce the repair bandwidth [13]. Theorem 8 holds for them unless some repetition code is used, because the problem can still be represented as P independent index coding problems. For example, Theorem 8 holds if a product matrix code [21] is used for distributed storage. However, a sub-optimal delivery latency is achieved, because each server stores a larger number of packets that have to be transmitted to the connected users for successful file reconstruction.*

To identify the acyclic sets $\mathcal{J}_p$ in the subgraph $G_p$, consider the permutations $\mathbf{u}=(u_1,\ldots,u_K)$ of $[K]$. To determine the tightest bound, we may only consider the largest such sets without a directed cycle. For a given $\mathbf{u}$, the largest set of nodes not containing a directed cycle is

$$\left\{C^p_{d_{u_i},\mathcal{A}_i} : \mathcal{A}_i \subseteq [1:K]\setminus\{u_1,\ldots,u_i\}, i=1,\ldots,K\right\}.$$

Each permutation $\mathbf{u}$ gives a unique acyclic set of nodes of the graph. The subsegment $C^p_{d_i,\phi}$ is not cached in any user, so there is no outgoing edge from $C^p_{d_i,\phi}$ to any other nodes in any sub-index coding problem. Therefore $C^p_{d_i,\phi}$ is always in the set $\mathcal{J}_p$.

Consider first $M_S=\frac{N}{\rho}$. In that case, $\sum_{j=1}^{\rho} n_{\pi(j)} r^{\pi(j)}_{d_i,\mathcal{A}} = \sum_{j=1}^{\rho} |C^{\pi(j)}_{d_i,\mathcal{A}}| = |W_{d_i,\mathcal{A}}|$. Following Theorem 8, in order to recover all the desired segments for each user, the deliver latency, $T_{sd}$ must satisfy

$$FT_{sd} \geq \sum_{p=1}^{P}\Bigg(\sum_{\mathcal{A}\subseteq[1:K]\setminus\{u_1\}}|C^p_{d_{u_1},\mathcal{A}}| + \cdots + \sum_{\mathcal{A}\subseteq[1:K]\setminus(\{u^i\}\cap\mathcal{K}_p)}|C^p_{d_{u_i},\mathcal{A}}| + \cdots + \sum_{\mathcal{A}\subseteq[1:K]\setminus(\{u^K\}\cap\mathcal{K}_p)}|C^p_{d_{u_K},\mathcal{A}}|\Bigg) \tag{36}$$

$$\text{s.t.} \quad \sum_{p\in\mathcal{Z}_j}|C^p_{d_{u_j},\mathcal{A}}| = |W_{d_{u_j},\mathcal{A}}| \qquad j\in[K], \tag{37}$$

for every permutation $\mathbf{u}$, and for every network topology.

We have $|C^p_{d,\mathcal{A}}| = \frac{|W_{d,\mathcal{A}}|}{\rho}$ for $M_S=\frac{N}{\rho}$, due to $(P,\rho)$ MDS coded storage. In Eq. (36), in the summation for a fixed value of $p$, the number of terms with $|\mathcal{A}|=i$ is $\binom{K}{i+1}-\binom{K-q_p}{i+1}$. Thus we have

$$FT_{sd} \geq \sum_{p=1}^{P}\sum_{i=0}^{K-1}\frac{\binom{K}{i+1}-\binom{K-q_p}{i+1}}{\binom{K}{i}}x^p_i \tag{38}$$

$$= \sum_{p=1}^{P}\sum_{i=0}^{K-1}\frac{\binom{K}{i+1}-\binom{K-q_p}{i+1}}{\rho\binom{K}{i}}x_i \tag{39}$$

$$= \sum_{i=0}^{K-1}\left[\sum_{p=1}^{P}\frac{\binom{K}{i+1}-\binom{K-q_p}{i+1}}{\rho\binom{K}{i}}\right]x_i \tag{40}$$

$$\text{while} \quad x_0+x_1+\cdots+x_K \geq F, \tag{41}$$

$$\text{and} \quad x_1+2x_2+\cdots+Kx_K \leq \frac{KM_U}{N}F \tag{42}$$

where $x_i \triangleq \sum_{\mathcal{A}\subset[K]:|\mathcal{A}|=i}|W_{j,\mathcal{A}}| = \binom{K}{i}|W_{j,\mathcal{A}}| = \rho\binom{K}{i}|C^p_{j,\mathcal{A}}|$ is the total normalized size of all segments of file $j$ cached by $i$ users; or equivalently, $x^p_i \triangleq \binom{K}{i}|C^p_{j,\mathcal{A}}| = \frac{1}{\rho}x_i$ is the total normalized size of all subsegments of file $j$ cached by $i$ users and stored in server $S_p$. We minimize the lower bound in Eq. (40) over all segment sizes $x_i$, which is a linear program with two linear constraints (41) and (42), where the former follows from the sum of all fractions of the files being one, while the latter follows from the user cache memory constraint. The solution of a linear program lies on one of the corner points of the feasible region. The feasible region defined by the constraints has only one corner point characterized by

$$x_i = \begin{cases} F & i=t, t=\frac{KM_U}{N} \\ 0 & \text{otherwise} \end{cases}.$$

Therefore, Eq. (40) simplifies to

$$T_{sd} \geq \sum_{p=1}^{P}\frac{\binom{K}{t+1}-\binom{K-q_p}{t+1}}{\rho\binom{K}{t}}, \tag{43}$$

which is achieved by our delivery scheme. This proves the optimality of the delivery scheme for successive transmission proposed in Section IV under the assumption of MDS coded storage at the servers and uncoded caching at the users.

### A. Redundancy in server storage

When there is redundant server storage capacity, i.e., server storage capacity is $M_S=\frac{N}{\rho-z}$, consider the constraint

$\sum_{j=1}^{\rho} n_{\pi(j)} r_{d_i,\mathcal{A}}^{\pi(j)} \geq |W_{d_i,\mathcal{A}}|$ in Theorem 8. Since the bound in Theorem 8 is a linear program of the rates of transmission of the coded subsegments from the servers, the optimal solution lies on one of the corner points of the feasible region defined by the constraint. The corner points for $M_S = \frac{N}{\rho - z}, z \in [\rho - 2]$, are those where $n_{\pi(j)} r_{d_i,\mathcal{A}}^{\pi(j)} = \frac{|W_{d_i,\mathcal{A}}|}{\rho - z}$ for all $j \in \mathcal{R}, \mathcal{R} \subset \mathcal{Z}_i, |\mathcal{R}| = \rho - z$, and equal to 0 for all $j \in \mathcal{Z}_i \setminus \mathcal{R}$. The optimal solution should lie on the corner point which chooses $\mathcal{R}$ such that the servers in $\mathcal{R}$ have the most multicasting opportunities, and can thus deliver $\rho - z$ coded subsegments of the requested segments to the users in the minimum number of transmissions. This is equivalent to finding the minimum cover for each multicast group as described in Section IV-A.

When fractional repetition (FR) codes are used for server storage [20], the minimum cover scheme may not be optimal. However, since FR codes have a maximum code rate of $\frac{1}{2}$, we cannot have distributed storage schemes where $M_S \leq \frac{2N}{P}$. Thus the minimum cover scheme is optimal for server storage capacities $M_S \leq \frac{2N}{P}$. We illustrate with a toy example that the bound in Theorem 8 does not hold when FR codes are used.

**Example 11.** *Consider the simple scenario with $P = 2$ servers, $K = 3$ users illustrated in Fig. 5, where we assume each server can store all the $N = 3$ files, i.e., $M_S = 3$, and each user has cache capacity $M_U = 1$. Let the cache contents of $U_1, U_2, U_3$ be $W_2, W_3, W_1$, respectively, and the demand vector $\mathbf{d} = \{W_1, W_2, W_3\}$. In this example, the demands can be satisfied by $S_1$ transmitting $W_1 \oplus W_2$, and $S_2$ transmitting $W_1 \oplus W_3$, that is, the delivery latency of $T_{sd} = 2$ is achievable. However, Theorem 8 gives the bound on delivery latency as $T_{sd} \geq 3$. $U_2$ receives its requested file $W_2$ with added interference of $W_1$ from $S_1$, which it cannot remove using its cache contents. However, $U_2$ adds the messages from $S_1$ and $S_2$ to align the interference on $W_2$ with $W_3$, which it can remove by using its cache contents, thus doing a sort of interference alignment. In contrast, if MDS coded storage were used, the interference alignment type of scheme would not be possible due to distinct coded subsegments transmitted by both servers.*

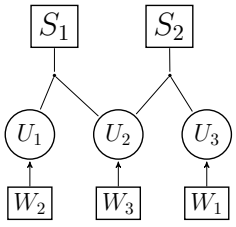



Fig. 5: Toy example with $P = 2, K = N = M_S = 3, M_U = 1$

The polymatroidal capacity region for multi-server index coding has been characterized in [18] for full user-server connectivity and uncoded server storage. Characterizing the capacity region for partial user-server connectivity, and constructing an optimal joint server storage and caching scheme for FR coded distributed storage is an interesting open problem for future work.

## VI. Parallel SBS Transmissions

When SBSs can deliver in parallel without interfering with each other, the normalized delivery latency is dictated by the SBS that has to deliver the maximum number of bits:

$$T_{pd} \triangleq \max_{q_p} \frac{1}{\rho \binom{K}{t}} \left[ \binom{K}{t+1} - \binom{K - q_p}{t+1} \right]. \quad (44)$$

The "best" and "worst" network topologies in the parallel transmission scenario are different from those in the successive transmission scenario. The most balanced topology, i.e., the one with the minimum value of the maximum $q_p$ has the "best" (lowest) delivery latency, contrary to the successive transmission scenario, in which this would be the "worst" topology. The corresponding delivery latency can be obtained by substituting $q_p = \lceil \frac{K\rho}{P} \rceil$ in (44). The topology with the maximum possible $q_p$, i.e., any topology with at least one server connected to all $K$ users, is the "worst" topology since it has the highest delivery latency.

### A. Redundant server storage capacity

The minimum server storage capacity that would allow the reconstruction of any demand combination is given by $M_S = \frac{N - M_U}{\rho}$. In this case, we cache the same $\frac{M_U}{N}$ fraction of the library in all the user caches during the placement phase, and deliver the remaining fraction of the demands from the servers without multicasting. The worst-case delivery latency in this case is $T_{pd} = \frac{K}{\rho}\left(1 - \frac{M_U}{N}\right)$.

Next, we consider the case when there is redundancy in server memories; that is, $\frac{N}{\rho} < M_S \leq N$. Assume that $M_S = \frac{N}{\rho - z}$ for some integer $z \in [\rho - 1]$. For non-integer values of $z$, the solution can be obtained by memory-sharing. Notice that, as for successive transmissions, users can select the servers from which to receive coded subsegments. A greedy server allocation algorithm is used. The algorithm assigns the multicast messages to the servers trying to keep the number of messages delivered by each server as evenly distributed as possible. At any point in time, if a server has delivered a higher number of messages than all the other servers, even if a better multicasting opportunity is available to this server, that server is not assigned a multicast message in order to balance the number of messages delivered by each server in a greedy manner. Instead, the server with the next best multicasting opportunity and a smaller count of transmissions is assigned to transmit a particular coded subsegment to a multicast group. Compare this with the algorithm for successive transmission, where a multicast message is always assigned to the server with the maximum multicasting opportunity. It is easy to see that the delivery latency achieved depends on the order in which the algorithm assigns multicast messages to the servers. Thus the proposed algorithm is suboptimal. Numerical results illustrating the performance of the proposed delivery algorithm will be presented in the next section.

## VII. Results and Discussions

In Fig. 6 we plot the achievable trade-off between the user cache capacity and the normalized delivery latency, $T_{sd}$, for

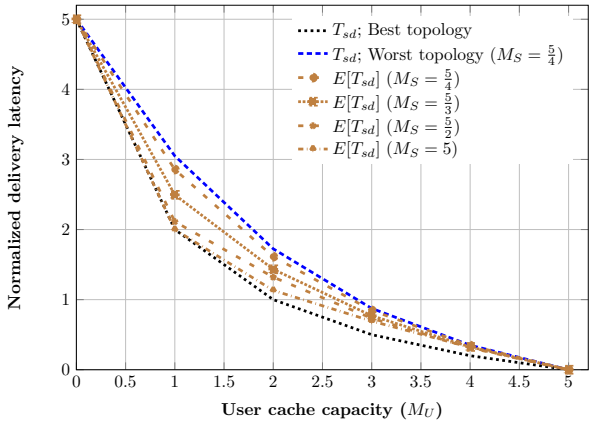



Fig. 6: Average normalized delivery latency vs. user cache capacity $M_U$, for $P = 7, N = K = 5, \rho = 4$, and for server storage capacities of $M_S = \frac{5}{4}, \frac{5}{3}, \frac{5}{2}, 5$.

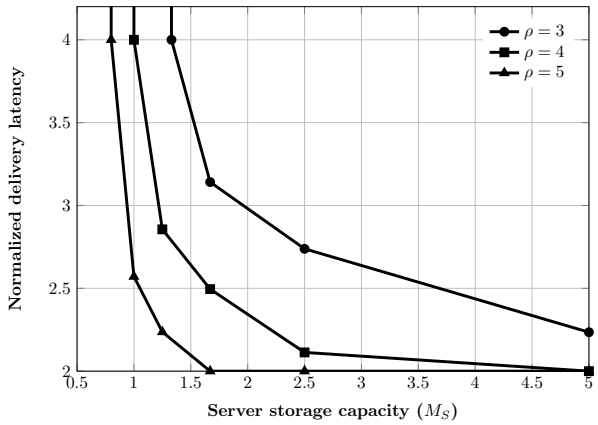



Fig. 7: Average normalized delivery latency vs. server storage capacity $M_S$, for $P = 7, N = K = 5, M_U = 1$ for successive SBS transmissions.

the best and worst topologies, and the average normalized delivery latency over all topologies, for successive transmission. The trade-off curves are plotted for different server storage capacities. We observe that the gap between the worst and the best topologies can be significant. From (17) and (23) we can deduce that, for successive transmission the worst topology delivery latency; and hence, the average delivery latency of the proposed scheme are both within a multiplicative factor of $\frac{1}{\alpha}$ of the best topology delivery latency. We observe from Fig. 6 that the delivery latency decreases significantly, particularly for low $M_U$ values, as the redundancy in server storage increases.

In Fig. 7 the average delivery latency for successive transmission is plotted as a function of the server storage capacity for server storage capacities $M_S \in [\frac{N-M_U}{\rho}, N]$. The figure is obtained by performing Monte Carlo simulations with uniform random realizations of the topology and averaging the delivery latency over them. We observe from Fig. 7 that the average delivery latency decreases rapidly for an initial increase in the server storage capacity, which is more significant for high $\rho$ values. This is because, thanks to MDS-coded storage at the servers, the number of available multicasting opportunities increases with the redundancy across servers. Fig. 7 highlights the fact that, for successive delivery and sufficient network connectivity, increasing the server storage beyond a certain value has little or no impact on the delivery latency.

In Fig. 8, it is shown that Eq. (34) in Section IV-B gives a fairly accurate estimate of the expected delivery latency for successive transmissions with redundant server storage capacity, especially for small server storage capacities. The theoretical estimate diverges a little from the expected rate for large server storage capacity, before again converging where the delivery latency saturates at the minimum. Also, comparing the plot for $\rho = 9, P = 21$ in Fig. 8 with the plot for $\rho = 3, P = 7$ in Fig. 7, where the connectivity $\alpha$ is the same, we observe that the average delivery latency decreases faster for $\rho = 9, P = 21$; that is, for larger values of $P$.

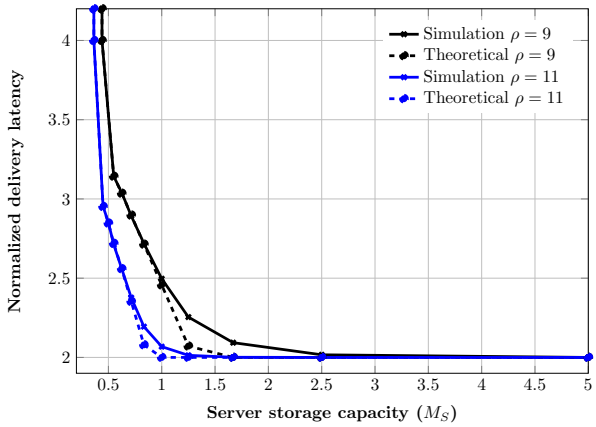



Fig. 8: Comparing the simulation with the theoretical, Average normalized delivery latency vs. server storage capacity, for $P = 21, N = K = 5, M_U = 1$ for successive SBS transmissions.

The average delivery latency for parallel transmissions is plotted with respect to the user cache capacity in Fig. 9, using (44). We observe as before that increasing the server storage capacity gives significant gains in the average delivery latency, especially for low values of $M_U$. Unlike the case for successive transmissions, the average delivery latency for $M_U = 0$ also

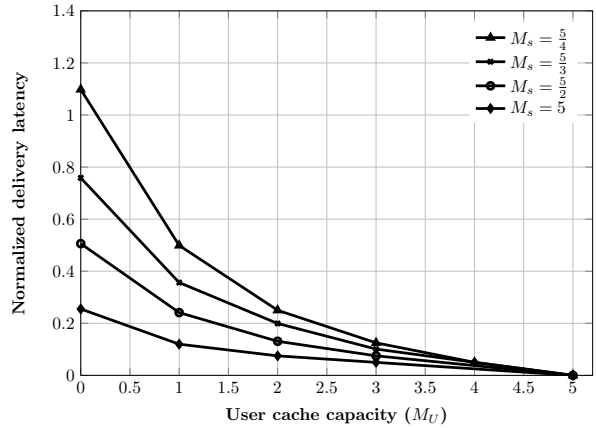



Fig. 9: Average normalized delivery latency for parallel transmissions vs. user cache capacity $M_U$, for $P = 7, N = K = 5, \rho = 4$, and for server storage capacities of $M_S = \frac{5}{4}, \frac{5}{3}, \frac{5}{2}, 5$.

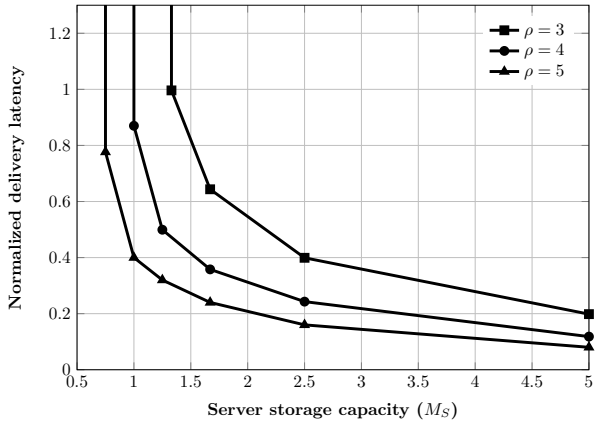



Fig. 10: Average normalized delivery latency vs. server storage capacity $M_S$, for $P = 7, N = K = 5, M_U = 1$ for parallel transmissions.

reduces as the server storage capacity is increased.

The average delivery latency for parallel transmissions is plotted with respect to the server storage capacity, $M_S$, in Fig. 10. Unlike the delivery latency for successive transmissions, we can see that the delivery latency does not saturate, and keeps decreasing until all the files are stored at each of the servers. We also observe as before that the increase in network connectivity $\alpha$ helps reduce the delivery latency significantly, especially for low server storage capacity $M_S$.

## VIII. Conclusions and future work

We have studied a multi-server coded caching and delivery network, in which cache-equipped users connect randomly to a subset of the available servers, each with its own limited storage capacity. While this allows each server to have only a limited amount of storage capacity, it requires coded storage across servers to account for the random topology. We proposed a joint coded storage, caching and delivery scheme that jointly applies MDS-coded storage at the servers, and uncoded caching and coded delivery to the users. The achievable delivery latency of this scheme for both successive and parallel transmissions from the SBSs are presented, with increasing user cache memory as well as increasing server storage capacity, and their averages over random network topologies are plotted. The analysis shows that when the server storage capacity is increased, the delivery latency can be reduced significantly, for both successive transmissions as well as parallel transmissions. However, it is also observed that for sufficient network connectivity, increasing server storage beyond a certain value provides little benefit. Increasing server storage has a more significant impact when there is low connectivity, and when user cache capacities are small.

An interesting open problem for future work is finding a lower bound and an optimal scheme when FR codes are used for distributed storage. A toy example 11 is given in this paper which illustrates the potential benefits of such codes. The toy example also presents an asymmetry in the user connections to the servers, where users 1 and 3 connect to one server each, while user 2 connects to 2 servers. An interesting problem is constructing a general scheme for heterogeneous network topologies and extracting gains from such topologies as demonstrated in the toy example. Another question relates to gains from heterogeneous distributed storage. For instance, if there is knowledge of user dynamics and non-uniform probabilities of the user-server connections, can a heterogeneous distributed storage scheme be designed to extract higher average gains? An extreme case of this scenario would mimic the combination network model where the user-server connections are completely fixed and known, which achieves higher gains. Such open problems present ripe material for future research.

## Acknowledgment

This work was supported in part by the European Union's H2020 research and innovation programme under the Marie Sklodowska-Curie Action SCAVENGE (grant agreement no. 675891), and by the European Research Council (ERC) Starting Grant BEACON (grant agreement no. 677854).

## References

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

[2] N. Mital, D. Gündüz and C. Ling, "Coded caching in a multi-server system with random topology," 2018 IEEE Wireless Communications and Networking Conference (WCNC), Barcelona, 2018, pp. 1-6.

[3] M. M. Amiri, Q. Yang and D. Gündüz, "Decentralized Caching and Coded Delivery With Distinct Cache Capacities," in IEEE Transactions on Communications, vol. 65, no. 11, pp. 4657-4669, Nov. 2017.

[4] K. Shanmugam, N. Golrezaei, A. G. Dimakis, A. F. Molisch and G. Caire, "FemtoCaching: Wireless Content Delivery Through Distributed Caching Helpers," in IEEE Transactions on Information Theory, vol. 59, no. 12, pp. 8402-8413, Dec. 2013.

[5] M. Franceschetti and R. Meester, "Random Networks for Communication: From Statistical Physics to Information Systems", Cambridge, Cambridge University Press, 2008.

[6] M. M. Amiri and D. Gündüz, "Fundamental Limits of Coded Caching: Improved Delivery Rate-Cache Capacity Tradeoff," in IEEE Transactions on Communications, vol. 65, no. 2, pp. 806-815, Feb. 2017.

[7] M. Gregori, J. Gómez-Vilardebó, J. Matamoros and D. Gündüz, "Wireless Content Caching for Small Cell and D2D Networks," in IEEE Journal on Selected Areas in Communications, vol. 34, no. 5, pp. 1222-1234, May 2016.

[8] J. Gómez-Vilardebó, "Fundamental Limits of Caching: Improved Rate-Memory Tradeoff With Coded Prefetching," in IEEE Transactions on Communications, vol. 66, no. 10, pp. 4488-4497, Oct. 2018.

[9] C. Tian and J. Chen, "Caching and Delivery via Interference Elimination," in IEEE Transactions on Information Theory, vol. 64, no. 3, pp. 1548-1560, March 2018.

[10] T. Luo, V. Aggarwal, and B. Peleato, "Coded caching with Distributed Storage", *ArXiv*:1611.06591v1 [cs.IT] Nov 2016.

[11] S. P. Shariatpanahi, S. A. Motahari and B. H. Khalaj, "Multi-Server Coded Caching," in IEEE Transactions on Information Theory, vol. 62, no. 12, pp. 7253-7271, Dec. 2016.

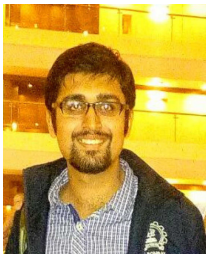
[12] M. Ji, A. M. Tulino, J. Llorca and G. Caire, "Caching in combination networks," 2015 49th Asilomar Conference on Signals, Systems and Computers, Pacific Grove, CA, 2015, pp. 1269-1273.

[13] A. G. Dimakis, P. B. Godfrey, Y. Wu, M. J. Wainwright and K. Ramchandran, "Network Coding for Distributed Storage Systems," in IEEE Transactions on Information Theory, vol. 56, no. 9, pp. 4539-4551, Sept. 2010.

[14] A. A. Zewail and A. Yener, "Coded caching for combination networks with cache-aided relays," 2017 IEEE International Symposium on Information Theory (ISIT), Aachen, 2017, pp. 2433-2437.

[15] K. Wan, D. Tuninetti and P. Piantanida, "On the optimality of uncoded cache placement," 2016 IEEE Information Theory Workshop (ITW), Cambridge, 2016, pp. 161-165.

[16] F. Arbabjolfaei, B. Bandemer, Y. Kim, E. Şaşoğlu and L. Wang, "On the capacity region for index coding," 2013 IEEE International Symposium on Information Theory, Istanbul, 2013, pp. 962-966.
[17] Z. Bar-Yossef, Y. Birk, T. S. Jayram and T. Kol, "Index Coding with Side Information," 2006 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS'06), Berkeley, CA, 2006, pp. 197-206.
[18] P. Sadeghi, F. Arbabjolfaei and Y. Kim, "Distributed index coding," 2016 IEEE Information Theory Workshop (ITW), Cambridge, 2016, pp. 330-334.
[19] M. Li, L. Ong and S. J. Johnson, "Cooperative Multi-Sender Index Coding," in IEEE Transactions on Information Theory, vol. 65, no. 3, pp. 1725-1739, March 2019.
[20] O. Olmez and A. Ramamoorthy, "Fractional Repetition Codes With Flexible Repair From Combinatorial Designs," in IEEE Transactions on Information Theory, vol. 62, no. 4, pp. 1565-1591, April 2016.
[21] K. V. Rashmi, N. B. Shah and P. V. Kumar, "Optimal Exact-Regenerating Codes for Distributed Storage at the MSR and MBR Points via a Product-Matrix Construction," in IEEE Transactions on Information Theory, vol. 57, no. 8, pp. 5227-5239, Aug. 2011.
[22] O. Olmez and A. Ramamoorthy, "Constructions of fractional repetition codes from combinatorial designs," 2013 Asilomar Conference on Signals, Systems and Computers, Pacific Grove, CA, 2013, pp. 647-651.



**Nitish Mital** received the B.Tech and M.Tech degrees in electrical engineering with specialization in communications and signal processing from Indian Institute of Technology Bombay in 2015. He secured an All India Rank of 309 in the prestigious Joint Entrance Exam for the Indian Institute of Technology. He is currently a research assistant and pursuing his PhD in Imperial College London. He was the recipient of the H2020 Marie-Sklodowska scholarship from 2016 to 2019. During his PhD, he visited the NYU Tandon School of Engineering as a Marie-Sklodowska early stage researcher in 2018.

His research interests lie in the areas of communications and information theory, distributed computing and security. He received the Best Student Paper Award at the 2018 IEEE Wireless Communications and Networking Conference (WCNC).

**Deniz Gündüz** [S03-M08-SM13] received the B.S. degree in electrical and electronics engineering from METU, Turkey in 2002, and the M.S. and Ph.D. degrees in electrical engineering from NYU Tandon School of Engineering (formerly Polytechnic University) in 2004 and 2007, respectively. After his PhD, he served as a postdoctoral research associate at Princeton University, and as a consulting assistant professor at Stanford University. He was a research associate at CTTC in Barcelona, Spain until September 2012, when he joined the Electrical and Electronic Engineering Department of Imperial College London, UK, where he is currently a Reader (Associate Professor) in information theory and communications, is the deputy head of the Intelligent Systems and Networks Group, 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. Dr. Gndz is the Area Editor (for Machine Learning and Communications) for the IEEE Transactions on Communications, and also serves as an Editor of the IEEE Transactions on Wireless Communications and IEEE Transactions on Green Communications and Networking. He is a Distinguished Lecturer for the IEEE Information Theory Society (2020-21). He is the recipient of the IEEE Communications Society - Communication Theory Technical Committee (CTTC) Early Achievement Award in 2017, a Starting Grant of the European Research Council (ERC) in 2016, IEEE Communications Society Best Young Researcher Award for the Europe, Middle East, and Africa Region in 2014, Best Paper Award at the 2019 IEEE Global Conference on Signal and Information Processing (GlobalSIP) and the 2016 IEEE Wireless Communications and Networking Conference (WCNC), and the Best Student Paper Awards at the 2018 IEEE Wireless Communications and Networking Conference (WCNC) and the 2007 IEEE International Symposium on Information Theory (ISIT).

**Cong Ling** is currently a Reader (equivalent to Professor/Associate Professor) in the Electrical and Electronic Engineering Department at Imperial College London. He is a member of the Academic Centre of Excellence in Cyber Security Research at Imperial College and an affiliated member of the Institute of Security Science and Technology of Imperial College.

He received the Bachelor and Master degrees from Nanjing Institute of Communications Engineering, China in 1995 and 1997 respectively, and the Ph.D. degree from Nanyang Technological University, Singapore in 2005. Before joining Imperial College, he had been on the faculties of Nanjing Institute of Communications Engineering and Kings College. He visited Hong Kong University of Science and Technology as a Hong Kong Telecom Institute of Information Technology (HKTIIT) Fellow in 2009.

Dr. Ling has been an Associate Editor (in multiterminal communications and lattice coding) of IEEE Transactions on Communications, and an Associate Editor of IEEE Transactions on Vehicular Technology and on the program committees of several international conferences including IEEE Information Theory Workshop, Globecom, and ICC. He is a member of IEEE.