Fundamental limits of caching: Improved rate-memory trade-off with coded prefetching

Jesús Gómez-Vilardebó, *Senior Member, IEEE,*

***Abstract*—We consider a cache network in which a single server is connected to multiple users via a shared error free link. The server has access to a database with N files of equal length F, and serves K users each with a cache memory of MF bits. A novel centralized coded caching scheme is proposed for scenarios with more users than files $N \leq K$ and cache capacities satisfying $\frac{1}{K} \leq M \leq \frac{N}{K}$. The proposed scheme outperforms the best rate-memory region known in the literature if $N \leq K \leq \frac{N^2+1}{2}$.**

***Index Terms*—Coded caching, network coding, index coding.**

I. Introduction

Content caching techniques are recently increasing attention to combat peak hour traffic in content delivery services. The basic idea is simple. If contents are made available at user terminals during low traffic periods, then the peak rate can be reduced. However, content requests are unknown to the server and thus content caching at user memories must be carefully chosen in order to be useful regardless of the contents requested during peak hours. The simplest caching scheme consists of storing each file partially at each user memory. Then, the server transmits the remaining requested data uncoded [1], [2]. For single user caching systems, this strategy is optimal. However, for multi-user systems, the seminal work in [3] by Maddah-Ali and Niesen shows that important gains can be obtained by a new coded caching strategy. Specifically, there authors show that, besides the local caching gain that is obtained by placing contents at user caches before they are requested, it is possible to obtain a global caching gain by creating broadcast opportunities. This is, by carefully choosing the content caches at different users, and using network coding techniques it is possible to transform the initial multi-cast network, where every user is requesting a different file, into a broadcast network, where every user requests exactly the same "coded" file, obtaining the new global caching gain.

The fundamental caching scheme developed in [3] was latter extended to more realistic situations. The decentralized setting was considered in [4], non-uniforms demands in [5]–[7], and online coded caching in [8], hierarchical cache network were considered in [9], [10], among others. In addition, new schemes pushing further the fundamental limits of caching systems have appeared in [11]–[19]. There have been also efforts to obtain theoretical lower bounds on the delivery rate. The cut-set bound was studied in [3]. A tighter lower bound was obtained in [20]. Through a computational approach a lower-bound for the special case $N = K = 3$ is derived in [21]. Other lower bounds have appeared in [13], [22], [23].

The work here proposed investigates the fundamental achievable rate for the particular situation where there are more users than files, and the caching memories at users are small compared to the number of files in the system. Besides its theoretical relevance, this situation can be readily found in the real world. For instance, global content delivery services such a Netflix serve a few multimedia contents to millions of users across the world. In addition, it was shown in [5] that a near optimal caching strategy consists in dividing the files into groups with similar popularity, and then applying the coded caching strategy to each group separately. Since the amount of users in each groups remains the same, when there are many groups, the cache size dedicated to each group is small as well as the number of files per user in each group.

The rest of this paper is organized as follows. Section II, presents the system model and the relevant previous works. Section III, summarizes the main results. Section IV describes the caching scheme proposed. In Section V an example is developed. Finally, conclusions are drawn in Section VI.

II. System Model and Previous Results

We consider a communication system with one server connected to K users, denoted as $U_1, ..., U_K$, through a shared, error-free link. There is a database at the server with N files, each of length F bits, denoted as $W_1, ..., W_N$. Each user is equipped with a local cache of capacity MF bits and is assumed to request only one full file. Here, we consider the special case where $M \in \left[0, \frac{N}{K}\right]$ and there are more users than files $N \leq K$. For convenience, we define parameter $q \triangleq \frac{N}{MK}$.

We consider the communication model introduced in [3]. The caching system operates in two phases: the *placement phase* and the *delivery phase*. In the placement phase, users have access to the server database, and each user fills his cache. As in [15], we allow coding in the prefetching phase. Then, each user U_k requests a single full file $W_{\mathbf{d}(k)}$ where $\mathbf{d} = (\mathbf{d}(1), ..., \mathbf{d}(K))$ denotes the demand vector. We denote the number of distinct requests in $\mathbf{d}$ as $N_e(\mathbf{d})$. In the delivery phase, only the server has access to the database. After being informed of the user demands, the server transmits a signal Y of size RF bits over the shared link to satisfy all user requests simultaneously. The signal Y is a function of the demand vector $\mathbf{d}$, all the files in the data base $W_1, ..., W_N$, and the content in the user caches $\boldsymbol{\mathcal{M}} = \{\mathcal{M}_1, ..., \mathcal{M}_K\}$. Using the local cache content and the received signal Y, each user U_k reconstructs its requested file $W_{\mathbf{d}(k)}$.

Let $\mathcal{D} = \{1, ..., N\}^K$, for a caching system (M, N, K), given a particular prefetching $\boldsymbol{\mathcal{M}}$ and a particular demand $\mathbf{d}$,

Manuscript received ; revised ; accepted; approved by IEEE Transactions on Communications Editor . This work was supported in part by the Catalan Government under Grant SGR2017-1479, and in part by the Spanish Government under Grant TEC2013-44591-P (INTENSYV). The author is with the Centre Tecnològic de Telecomunicacions de Catalunya (CTTC/CERCA), 08860 Barcelona, Spain (e-mail: jesus.gomez@cttc.cat).

we say that communication rate R is achievable if and only if there exists a message Y of length RF bits such that every user U_k is able to reconstruct its desired file $W_{\mathbf{d}(k)}$. For a particular prefetching $\boldsymbol{\mathcal{M}}$ and demand $\mathbf{d}$, we denote the achievable rate as $R(\mathbf{d}, \mathcal{M})$. Then, the rate needed for the worst demand is given by $R^*(\boldsymbol{\mathcal{M}}) = \max_{\mathbf{d}\in\mathcal{D}} R(\mathbf{d}, \boldsymbol{\mathcal{M}})$. Finally, we define the rate-memory pair (R^*, M), or the rate-memory trade-off function $R^*(M)$ as the minim rate R^* for different memory constraints M, i.e. we aim to find $R^* = \min_{\boldsymbol{\mathcal{M}}} R^*(\boldsymbol{\mathcal{M}})$ where the minimization is over all caching schemes $\boldsymbol{\mathcal{M}}$ satisfying the memory load constrain M. Observe, that if users have no caching capacity $M = 0$, the server needs to send the full requested files and thus, the worst demand rate is $R^* = N$. Instead, if users can have a complete copy of the server's database $M = N$, then no information needs to be transmitted from the server $R^* = 0$.

A. *Previous Results*

For the special case considered here $M \in \left[0, \frac{N}{K}\right]$ and $N \leq K$, the best known rate-memory function in the literature can be obtained by memory sharing between four achievable rate-memory pairs: the trivial rate-memory pair $(N, 0)$, the rate-memory pair obtained in [15] for $M = \frac{1}{K}$

$$(R^*_{\text{CFL}}, M_{\text{CFL}}) = \left(N - \frac{N}{K}, \frac{1}{K}\right), \tag{1}$$

the rate-memory pair obtained by the schemes proposed in [11] and [16] for $M = \frac{N}{K}$

$$(R^*_{\text{GBC}}, M_{\text{GBC}}) = \left(N - \frac{N(N+1)}{2K}, \frac{N}{K}\right), \tag{2}$$

and the rate-memory pairs obtained in [17]

$$(R^*_{\text{MDS}}, M_{\text{MDS}}) = \left(\frac{N\,(K-t)}{K}, \frac{t\,[(N-1)t + K - N]}{K\,(K-1)}\right), \tag{3}$$

with $t = 0, 1, ..., K$. The lower convex envelope of all these rate-memory pairs, provides the best rate-memory function in the literature. The scheme in [15] makes use of coded prefetching at users' cache. As shown in [15], the scheme achieving (1) is optimal for $M = \frac{1}{K}$. The schemes proposed in [11] and [16] assume uncoded prefetching. They are essentially the same at $M = \frac{N}{K}$. The scheme proposed in [16] was shown to be optimal among all the uncoded prefetching schemes. The design of the scheme here proposed was initially motivated to connect the schemes achieving the rate memory-pairs in (1) and (2), beyond the simple memory sharing between both rate-memory points. To achieve the rate memory pair in (2), the strategy described in [11] and [16] divides each of the N files into K subfiles of equal size, and stores each subfile in a different user, requiring a cache load of $M = \frac{N}{K}$. Using the same subfile partition scheme, but storing only the XOR of the N subfiles at each user, and thus reducing content cached to one coded cached subfile per user i.e. $M = \frac{1}{K}$, it was shown in [15] that the optimal rate-memory pair in (1) can be achieved. The strategy here proposed aims at extending the idea of coding together subfiles of different files at each user, in order to find intermediate rate-memory points between (1) and (2). To that end, instead of caching the XOR of all N subfiles, we XOR $q \in \{1, ..., N\}$ different subfiles. To keep the system symmetry in the caches, we store a different coded subfile for each of the $\binom{N}{q}$ possible combinations of q subfiles at each user. We show that to construct all these coded subfiles, it is sufficient to split each file into $F_q = K\binom{N-1}{q-1}$ subfiles. As a result, the memory load at each user cache is $M = \binom{N}{q}/K\binom{N-1}{q-1} = \frac{N}{Kq}$. As we show later in Corollary 1.1, if $K \geq N$ then the rate-memory points obtained by the caching scheme proposed is

$$(R^*, M) = \left(N - \frac{N+1}{q+1}\frac{N}{K}, \frac{N}{Kq}\right) \tag{4}$$

for $q \in \{1, ..., N\}$. Observe that by memory-sharing between (2), and (1), we obtain the rate-memory function

$$R_{\text{CFL/GBC}}(M) = N - \frac{N}{2K} - \frac{N}{2}M$$

which particularizing to the memory loads $M = \frac{N}{Kq}$, returns

$$R_{\text{CFL/GBC}}\left(\frac{N}{Kq}\right) = N - \frac{N+q}{2q}\frac{N}{K} \tag{5}$$

By comparing (4) and (5), we have that our scheme strictly outperforms the memory sharing between the CFL and GBC rate-memory points every where in $0 < M < \frac{N}{K}$. The price to pay for these rate reduction is a much higher subpacketization requirement. Observe that the CFL $(q = N)$ and GBC $(q = 1)$ rate-memory pairs require $F_1 = F_N = K$ subfile partitions. Thus, to obtain the rates in (5) for any $q \in \{2, ..., N-1\}$ via memory-sharing between these two schemes, we need to slip each file into $F_{1-N} = F_1 + F_N = 2K$ subfiles. Instead, our scheme requires $F_q = K\binom{N-1}{q-1}$, which for $q = \frac{N+1}{2}$ coincides with the Catalan number $F_{\frac{N-1}{2}} = C_{\frac{N-1}{2}}$ and thus, asymptotically grows as $K\frac{2^{N-1}}{\left(\frac{N-1}{2}\right)^{3/2}\sqrt{\pi}}$, i.e. exponentially with the number of files N.

Finally, the scheme developed in [17] for situations with more users than files $K \geq N$ makes use of binary codes, in particular maximum distant separable (MDS) codes and rank metric codes to obtain the rate-memory pairs in (3), which are shown to be optimal at certain points. For small cache memories $M \leq \frac{N}{K}$, our scheme improves [17] for a high-moderate number of users, i.e. $K \leq \frac{N^2+1}{2}$. A method to obtain new rate-memory points is described in [18]. However no explicit characterization of these rate-memory pairs is given.

There have been other coded prefetching schemes proposed, see [14] and [12] but either they do not improve the current best known rate-memory trade-off or they apply to other situations. The optimal rate-memory trade-off for a caching systems remains an open problem. Besides the achievable rate-memory trade-off described above, there have been efforts to obtain theoretical lower bounds on the delivery rate. The cut-set bound was studied in [3]. A tighter lower bound was obtained in [20]. Through a computational approach a lower-bound for the special case $N = K = 3$ is derived in [21]. Other lower bounds have appeared in [13], [22], [23].

III. Main Result

The following theorem presents the delivery rate obtained by the proposed caching scheme for a particular demand $\mathbf{d}$.

Theorem 1. *For a caching problem with K users and N files, local cache size of M files at each user, and parameter $q = \frac{N}{MK}$. Given a particular demand $\mathbf{d}$, let $N_e(\mathbf{d})$ be the number of distinct file requests, then the delivery rate*

$$R = \frac{K N_e(\mathbf{d}) \binom{N-1}{q-1} - N_e(\mathbf{d}) \left(\frac{N_e(\mathbf{d})+1}{q+1} \right) \binom{N_e(\mathbf{d})-1}{q-1}}{K \binom{N-1}{q-1}} \tag{6}$$

is achievable for $q \in \{1, ..., N\}$. Furthermore, for $q \in [1, N]$ the rate-memory pairs in the lower convex envelope of its values at $q \in \{1, ..., N\}$ are achievable.

We prove this result in the following section by describing the new caching scheme. The delivery rate presented in Theorem 1 is valid for any K and N, however, it is particularly useful for $K \geq N$, as we detail in the next corollaries. The next corollary, establishes the achievable rate memory pair for the worst case demand. The proof is provided in the Appendix.

Corollary 1.1. *For a caching problem with K users and N files, local cache size of M files at each user, and parameter $q = \frac{N}{MK}$, the delivery rate-memory pairs (R^*, M)*

$$(R^*, M) = \left(\bar{N} - \frac{N}{K} \frac{\bar{N}+1}{q+1} \frac{\binom{\bar{N}}{q}}{\binom{N}{q}}, \frac{N}{Kq} \right) \tag{7}$$

with $\bar{N} = \min(N, K)$ are achievable for $q \in \{1, ..., N\}$. Furthermore, for $q \in [1, N]$ the rate-memory pairs in the lower convex envelope of its values at $q \in \{1, ..., N\}$ are achievable.

Remark 1.1. *The rate-memory function in (7) coincides with the rate-memory function for the CFL scheme in (1) for $q = N$ and with the one for the GBC scheme in (2) for $q = 1$. For $q = 1$, $M = \frac{N}{K}$ our scheme is essentially the same as the one described in [16] and [11]. However, the scheme proposed in [15] to achieve (1) differs slightly from the one considered here for $K > N$. Indeed, while [15] divides each file into NK subfiles, our scheme requires only K subfiles per file.*

The next corollaries compare the proposed scheme with the scheme presented in [16], the scheme in [17], the cut set bound derived in [3], and the outer bound in [20]. These results are proved in the Appendices. First, we show that the scheme proposed here improves the state of the art also if $N \geq K$.

Corollary 1.2. *If $N \geq K$ and $M \leq \frac{N}{K}$, the best known rate-memory function is $R^*_{YU}(M) = K - \frac{K}{N}\frac{K+1}{2}M$, see [16]. By evaluating the rate-memory pairs (R^*, M) in (7) at $M = \frac{N}{Kq}$ for $q = 1, ..., N$, we have that our scheme outperforms [16], if $q \leq K$ and $\frac{2q}{q+1} \geq \prod_{i=1}^{q-1} \left(\frac{N-i}{K-i} \right)$. In particular, for $q = 2$, we require $\frac{4K-1}{3} \geq N \geq K$.*

Corollary 1.3. *For a caching problem with K users and N files, $K \geq N$, and local cache size of $\frac{1}{K} \leq M \leq \frac{N}{K}$, a sufficient condition for the proposed scheme to outperform the rate-memory region obtained by the lower convex envelope of the rate-memory pairs in (3), is $K \leq \frac{N^2+1}{2}$.*

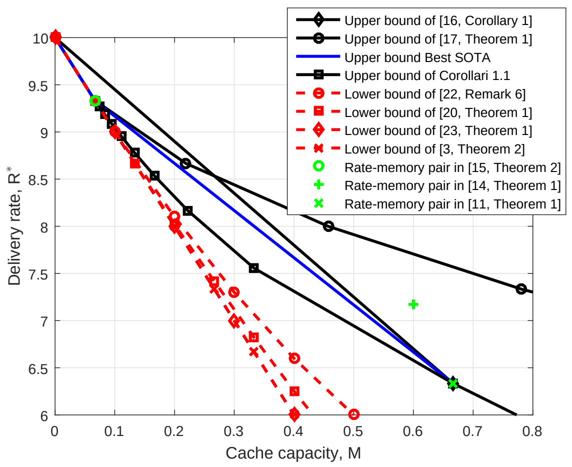

Fig. 1: Rate-memory functions of the proposed scheme in Corollary 1.1 compared with existing schemes and lower bounds in the literature for $N = 10$ and $K = 15$.

Corollary 1.4. *Let $R_{CB}(M)$ denote the rate-memory function obtained for the cut set bound. The rate difference between the cut set rate and the rate-memory pairs in Corollary 1.1 is*

$$R_{CB}(M) - R^*(M) = \frac{N}{K} \left(\frac{N}{q} - \frac{N+1}{q+1} \right).$$

Observe that, as first reported in [15, Theorems 3 and 4], for $q = N$, $M = \frac{1}{K}$ the cut set lower bound is achieved by the proposed strategy.

Corollary 1.5. *Let $R_{STC}(M)$ denote the outer bound on the rate-memory function presented in [20]. For $K = N$ and $M = \frac{N}{(N-1)K} = \frac{1}{N-1}$, this bound is achievable by the rate-memory function in Corollary 1.1. This is $R^* \left(\frac{1}{N-1} \right) = R_{STC} \left(\frac{1}{N-1} \right)$.*

We conclude this section by illustrating in Fig. 1 the worst demand rate-memory function for the proposed scheme Corollary 1.1 and for the state of the art (SOTA). We consider the case $N = 10$ files and $K = 15$ users. We provide the rate-memory regions in [16, Corollary 1], [17, Theorem 1], the rate memory pairs in [15, Theorem 2], [14, Theorem 1], and [11, Theorem 1]. We also include in this figure for comparison, the cut set lower bound [3, Theorem 2], the information-theoretical lower bound obtained in [20, Theorem 1], and the lower bounds recently appeared in [23, Theorem 1] and [22, Remark 6]. We observe that, for the special situation considered here, the new proposed scheme obtains a significant improvement with respect to the previous best SOTA.

IV. Proposed Caching Scheme

In this section, we describe the caching scheme proposed. We provide an example in the next section. Let us define the set of user indexes as $\mathcal{K} = \{1, ..., K\}$, and the set of file indexes as $\mathcal{F} = \{1, ..., N\}$. Consider a cache capacity at users of $M = \frac{N}{qK}$. To achieve the rate R stated in Theorem 1, we present a prefetching and delivery scheme for $q \in \{1, ..., N\}$, since for general $\frac{1}{K} \leq M \leq \frac{N}{K}$, the minimum rate can be achieved by memory sharing.

q	M	**Prefetching User** U_i
1	$\frac{1}{2}$	$W_1^{(i)}$,$W_2^{(i)}$,$W_3^{(i)}$
2	$\frac{1}{4}$	$W_{1,\{1,2\}}^{(i)} \oplus W_{2,\{1,2\}}^{(i)}$ $W_{1,\{1,3\}}^{(i)} \oplus W_{3,\{1,3\}}^{(i)}$ $W_{2,\{2,3\}}^{(i)} \oplus W_{3,\{2,3\}}^{(i)}$
3	$\frac{1}{6}$	$W_{1,\{1,2,3\}}^{(i)} \oplus W_{2,\{1,2,3\}}^{(i)} \oplus W_{3,\{1,2,3\}}^{(i)}$

TABLE I: Prefetching schemes at user U_i for the proposed coded caching scheme when $K = 6$, $N = 3$, and $q \in \{1, 2, 3\}$.

$\mathcal{T}(q) = \{(i, f, \mathcal{A}) :$ for all $i \in \mathcal{K},\ \mathcal{A} \subseteq \mathcal{F}, \lvert\mathcal{A}\rvert = q$ and $f \in \mathcal{A}\}$
$\mathcal{R}(q, \mathbf{d}) = \{(i, f, \mathcal{A}) \in \mathcal{T}(q) :$ for all $f \in \mathcal{N}_e(\mathbf{d})\}$
$\mathcal{R}_I(q, \mathbf{d}) = \{(i, f, \mathcal{A}) \in \mathcal{R}(q, \mathbf{d}) : \mathcal{A} \not\subseteq \mathcal{N}_e(\mathbf{d})\}$
$\mathcal{R}_{II}(q, \mathbf{d}) = \{(i, f, \mathcal{A}) \in \mathcal{R}(q, \mathbf{d}) :\ \mathcal{A} \subseteq \mathcal{N}_e(\mathbf{d}), \mathbf{d}(i) \in \mathcal{A}\}$
$\mathcal{R}_{III}(q, \mathbf{d}) =\{(i, f, \mathcal{A}) \in \mathcal{R}(q, \mathbf{d}) :\ \mathcal{A} \subseteq \mathcal{N}_e(\mathbf{d}), \mathbf{d}(i) \notin \mathcal{A}\}$

TABLE II: 3-tuple sets identifying subfiles $W_{f,\mathcal{A}}^{(i)}$.

Prefetching scheme: Given $q \in \{1, ..., N\}$, consider the $\binom{N}{q}$ possible subsets $\mathcal{A} \subseteq \mathcal{F}$ of q different files, $|\mathcal{A}| = q$. First, each file W_f is partitioned into K parts, $W_f^{(i)}$, one for each user $i \in \mathcal{K}$. Then, each of these parts is further partitioned into $\binom{N-1}{q-1}$ subfiles $W_{f,\mathcal{A}}^{(i)}$, one for each subset $\mathcal{A}$ of q files that satisfies $f \in \mathcal{A}$. Thus, we broke each file into a total of $K\binom{N-1}{q-1}$ subfiles. Finally, user U_i computes and stores the *coded cached subfiles*

$$Z_{\mathcal{A}}^{(i)} = \bigoplus_{f\in\mathcal{A}} W_{f,\mathcal{A}}^{(i)}$$

for all subsets $\mathcal{A} \in \{\mathcal{A} \subseteq\mathcal{F} : |\mathcal{A}| = q\}$. Because there are $\binom{N}{q}$ subsets $\mathcal{A}$ and each subfile has $F/K\binom{N-1}{q-1}$ bits, the required cache load at each user equals $MF = \binom{N}{q}F/K\binom{N-1}{q-1} = \frac{N}{qK}F$ bits. The coded cached subfiles at user U_i for a scenario with $N = 3$ and $q \in \{1, ..., 3\}$ are detailed in Table I.

Delivery scheme: Consider the sets defined in Table II. Each subfile $W_{f,\mathcal{A}}^{(i)}$ satisfies $f \in \mathcal{A}$ and is only XORed in one coded cached subfile $Z_{\mathcal{A}}^{(i)}$. Thus, each subfile $W_{f,\mathcal{A}}^{(i)}$ can be identified by a 3-tuple $(i, f, \mathcal{A}) \in \mathcal{T}(q)$. Let $\mathcal{N}_e(\mathbf{d})$ be the set of requested files. Then, the subfiles requested by some user satisfy $f \in \mathcal{N}_e(\mathbf{d})$, and are thus, identified by the 3-tuples $(i, f, \mathcal{A}) \in \mathcal{R}(q, \mathbf{d})$. The delivery scheme proposed here divides the requested subfiles into three types, and obtains the requested subfiles in each type, separately. The requested subfiles Type I, $(i, f, \mathcal{A}) \in \mathcal{R}_I(q, \mathbf{d})$, are those requested subfiles, which are coded in the coded cached subfiles $Z_{\mathcal{A}}^{(i)}$ together with at least one subfile not requested by any user $\mathcal{A} \not\subseteq \mathcal{N}_e(\mathbf{d})$. The requested subfiles Type II, $(i, f, \mathcal{A}) \in \mathcal{R}_{II}(q, \mathbf{d})$, include all the requested subfiles coded in the coded cached subfiles $Z_{\mathcal{A}}^{(i)}$ together with other requested files $\mathcal{A} \subseteq N_e(\mathbf{d})$ and placed in the cache of a user U_i requesting one of them, i.e. $\mathbf{d}(i) \in \mathcal{A}$. Finally, the requested subfiles Type III, $(i, f, \mathcal{A}) \in \mathcal{R}_{III}(q, \mathbf{d})$, include all the requested subfiles coded in the coded cached subfiles $Z_{\mathcal{A}}^{(i)}$ together with other also requested files $\mathcal{A} \subseteq N_e(\mathbf{d})$, different from the file requested by the user caching them, i.e. $\mathbf{d}(i) \notin \mathcal{A}$.

Delivery of requested subfiles Type I: For these subfiles, the server simply broadcasts them one by one, i.e. $Y_{f,\mathcal{A}}^{(i)} = W_{f,\mathcal{A}}^{(i)}$ for all $(i, f, \mathcal{A}) \in \mathcal{R}_I(q, \mathbf{d})$. Given that

$$\mathcal{R}_I(q, \mathbf{d}) = \mathcal{T}(q, \mathbf{d})\backslash \{(i, f, \mathcal{A}) \in \mathcal{R}(q, \mathbf{d}) : \mathcal{A} \subseteq \mathcal{N}_e(\mathbf{d})\}.$$

The total number of requested subfiles Type I, and thus of broadcasted subfiles Type I is

$$\begin{aligned} T_I &= \sum_{(i,f,\mathcal{A})\in\mathcal{T}(q,\mathbf{d})} 1 - \sum_{(i,f,\mathcal{A})\in\mathcal{R}(q,\mathbf{d}):\mathcal{A}\subseteq\mathcal{N}_e(\mathbf{d})} 1 \\ &= \sum_{i\in\mathcal{K}} \sum_{f\in\mathcal{N}_e(\mathbf{d})} \left(\sum_{\mathcal{A}:f\in\mathcal{A}} 1 - \sum_{\mathcal{A}:\{f\in\mathcal{A},\mathcal{A}\subseteq\mathcal{N}_e(\mathbf{d})\}} 1 \right) \\ &= KN_e(\mathbf{d}) \left(\binom{N-1}{q-1} - \binom{N_e(\mathbf{d})-1}{q-1} \right). \quad (8) \end{aligned}$$

Although not explicit written for brevity, we require $\mathcal{A} \subseteq \mathcal{F}$ and $|\mathcal{A}| = q$. Equality (8) follows since, from right to left, first, the number of sets $\mathcal{A} \subseteq \mathcal{F}$, with $|\mathcal{A}| = q$ that include a particular file $f \in \mathcal{A}$ is $\binom{N-1}{q-1}$, whereas the number of sets satisfying $\mathcal{A} \subseteq \mathcal{N}_e(\mathbf{d})$ that include a particular file $f \in \mathcal{A}$ is $\binom{N_e(\mathbf{d})-1}{q-1}$, second, there are $N_e(\mathbf{d})$ files in $\mathcal{N}_e(\mathbf{d})$ and, third, there are K users in $\mathcal{K}$.

Delivery of requested subfiles Type II: First, the server arbitrarily selects one user *leader* $u_f \in \mathcal{K}(f)$ for each file $f \in \mathcal{F}$. Let $\bar{\mathcal{K}}(f)$ denote the set of users requesting a file different from W_f. Then, the server broadcasts

$$Y_{f,\mathcal{A}}^{(i)} = \begin{cases} W_{f,\mathcal{A}}^{(i)} & \text{if } i \in \bar{\mathcal{K}}(f) \\ W_{f,\mathcal{A}}^{(i)} \oplus W_{f,\mathcal{A}}^{(u_f)} & \text{if } i \in \mathcal{K}(f)\backslash u_f \end{cases} \quad (9)$$

for all $(i, f, \mathcal{A}) \in \mathcal{R}_{II}(q, \mathbf{d})$ satisfying $i\in \mathcal{K}\backslash u_f$.

The rationale for this broadcasting strategy is the following. A user U_k requesting file $W_{\mathbf{d}(k)}$ obtains the subfiles Type II, $W_{\mathbf{d}(k),\mathcal{A}}^{(i)}$, coded cached at users requesting a different file $i \in \bar{\mathcal{K}}(\mathbf{d}(k))$, directly, from $Y_{\mathbf{d}(k),\mathcal{A}}^{(i)} = W_{\mathbf{d}(k),\mathcal{A}}^{(i)}$. Next, user U_k obtains the subfile Type II $W_{\mathbf{d}(k),\mathcal{A}}^{(k)}$ coded cached at $Z_{\mathcal{A}}^{(k)}$ in its own cache for all sets $\mathcal{A}$ with $\mathbf{d}(k) \in \mathcal{A}$. To that end, user U_k XORs to $Z_{\mathcal{A}}^{(k)}$ all the broadcasted subfiles $Y_{f,\mathcal{A}}^{(k)} = W_{f,\mathcal{A}}^{(k)}$ for all files $f \in \mathcal{A}\backslash\mathbf{d}(k)$. After all users requesting a particular file, $W_{\mathbf{d}(k)}$, obtain the subfiles Type II coded in their own cache, $\mathrm{W}_{\mathbf{d}(k),\mathcal{A}}^{(k)}$, they make use of the broadcasted subfiles $Y_{\mathbf{d}(k),\mathcal{A}}^{(i)} = W_{\mathbf{d}(k),\mathcal{A}}^{(i)} \oplus W_{\mathbf{d}(k),\mathcal{A}}^{(u_{\mathbf{d}(k)})}$ for all $i \in \mathcal{K}(\mathbf{d}(k))\backslash u_{\mathbf{d}(k)}$ to exchange them. Observe that, the user leader already has $W_{\mathbf{d}(k),\mathcal{A}}^{(u_{\mathbf{d}(k)})}$ since it is coded in its cache, and thus can use $Y_{\mathbf{d}(k),\mathcal{A}}^{(i)} = W_{\mathbf{d}(k),\mathcal{A}}^{(i)} \oplus W_{\mathbf{d}(k),\mathcal{A}}^{(u_{\mathbf{d}(k)})}$ for all $i \in \mathcal{K}(\mathbf{d}(k))\backslash u_{\mathbf{d}(k)}$ to obtain the remaining subfiles. Similarly, not user leaders, $k \in \mathcal{K}(\mathbf{d}(k))\backslash u_{\mathbf{d}(k)}$, already have $W_{\mathbf{d}(k),\mathcal{A}}^{(k)}$ and thus, can obtain the subfile cached at the user leader $W_{\mathbf{d}(k),\mathcal{A}}^{(u_{\mathbf{d}(k)})}$ from $Y_{\mathbf{d}(k),\mathcal{A}}^{(k)} = W_{\mathbf{d}(k),\mathcal{A}}^{(k)} \oplus W_{\mathbf{d}(k),\mathcal{A}}^{(u_{\mathbf{d}(k)})}$, and then the rest from $Y_{\mathbf{d}(k),\mathcal{A}}^{(i)} = W_{\mathbf{d}(k),\mathcal{A}}^{(i)} \oplus W_{\mathbf{d}(k),\mathcal{A}}^{(u_{\mathbf{d}(k)})}$ for all if $i \in \mathcal{K}(\mathbf{d}(k))\backslash\{u_{\mathbf{d}(k)}, k\}$.

Next, we detail the decoding of subfiles Type II at user U_k. This user requests file $W_{\mathbf{d}(k)}$ and, thus, is only interested in the subfiles $W_{\mathbf{d}(k),\mathcal{A}}^{(i)}$ for all 2-tuples $(i, \mathcal{A})$ such that $(i, \mathbf{d}(k), \mathcal{A}) \in \mathcal{R}_{II}(q, \mathbf{d})$. The decoding process at user k begins by computing

$$W_{\mathbf{d}(k),\mathcal{A}}^{(i)} = Z_{\mathcal{A}}^{(i)} \oplus \bigoplus_{f\in\mathcal{A}\backslash\mathbf{d}(k)} Y_{f,\mathcal{A}}^{(i)} \quad (10)$$

for all $(i,\mathcal{A})$ such that $i=k$ and $(i,\mathbf{d}(k),\mathcal{A})\in\mathcal{R}_{II}(q,\mathbf{d})$, and then $W^{(i)}_{\mathbf{d}(k),\mathcal{A}}$

$$= \begin{cases} Y^{(i)}_{\mathbf{d}(k),\mathcal{A}} & \text{if } i\in\bar{\mathcal{K}}(\mathbf{d}(k)) \\ Y^{(i)}_{\mathbf{d}(k),\mathcal{A}}\oplus W^{(k)}_{\mathbf{d}(k),\mathcal{A}} & \text{if } i\in\mathcal{K}(\mathbf{d}(k)) \text{ and } k=u_{\mathbf{d}(k)} \\ Y^{(k)}_{\mathbf{d}(k),\mathcal{A}}\oplus W^{(k)}_{\mathbf{d}(k),\mathcal{A}} & \text{if } i=u_{\mathbf{d}(k)} \text{ and } k\neq u_{\mathbf{d}(k)} \\ Y^{(i)}_{\mathbf{d}(k),\mathcal{A}}\oplus Y^{(k)}_{\mathbf{d}(k),\mathcal{A}} \oplus W^{(k)}_{\mathbf{d}(k),\mathcal{A}} & \text{if } i\in\mathcal{K}(\mathbf{d}(k))\backslash u_{\mathbf{d}(k)}, \text{ and } k\neq u_{\mathbf{d}(k)} \end{cases} \tag{11}$$

for all 2-tuples $(i,\mathcal{A})$ such that $i\neq k$ and $(i,\mathbf{d}(k),\mathcal{A})\in\mathcal{R}_{II}(q,\mathbf{d})$. To show this result, observe that for any 2-tuple $(f,\mathcal{A})$ such that $(k,f,\mathcal{A})\in\mathcal{R}_{II}(q,\mathbf{d})$ and $f\in\mathcal{A}\backslash\mathbf{d}(k)$, we have $\mathcal{A}\backslash\mathbf{d}(k)\subseteq\bar{\mathcal{K}}(\mathbf{d}(k))$ and thus, $Y^{(k)}_{f,\mathcal{A}}=W^{(k)}_{f,\mathcal{A}}$ from (9). Using the coded cached subfiles, $Z^{(k)}_{\mathcal{A}}$, user U_k can compute

$$\begin{aligned} W^{(i)}_{\mathbf{d}(k),\mathcal{A}} &= Z^{(k)}_{\mathcal{A}}\oplus\bigoplus_{f\in\mathcal{A}\backslash\mathbf{d}(k)} Y^{(k)}_{f,\mathcal{A}} \\ &= \bigoplus_{f\in\mathcal{A}} W^{(k)}_{f,\mathcal{A}}\oplus\bigoplus_{f\in\mathcal{A}\backslash\mathbf{d}(k)} W^{(k)}_{f,\mathcal{A}} \end{aligned} \tag{12}$$

for all $(i,\mathcal{A})$ such that $i=k$ and $(i,\mathbf{d}(k),\mathcal{A})\in\mathcal{R}_{II}(q,\mathbf{d})$.

Next, consider the decoding of subfiles $W^{(i)}_{\mathbf{d}(k),\mathcal{A}}$ in (11) for all $(i,\mathcal{A})$ such that $i\neq k$ and $(i,\mathbf{d}(k),\mathcal{A})\in\mathcal{R}_{II}(q,\mathbf{d})$. User U_k can obtain, directly, from the broadcasted subfiles Type II in (9), the subfiles $W^{(i)}_{\mathbf{d}(k),\mathcal{A}}=Y^{(i)}_{\mathbf{d}(k),\mathcal{A}}$ for all $i\in\bar{\mathcal{K}}(\mathbf{d}(k))$. Next, if $k=u_{\mathbf{d}(k)}$, using $W^{(k)}_{\mathbf{d}(k),\mathcal{A}}$ from (12), he obtains

$$\begin{aligned} W^{(i)}_{\mathbf{d}(k),\mathcal{A}} &= Y^{(i)}_{\mathbf{d}(k),\mathcal{A}}\oplus W^{(k)}_{\mathbf{d}(k),\mathcal{A}} \\ &= W^{(i)}_{\mathbf{d}(k),\mathcal{A}}\oplus W^{(u_{\mathbf{d}(k)})}_{\mathbf{d}(k),\mathcal{A}}\oplus W^{(k)}_{\mathbf{d}(k),\mathcal{A}} \\ &= W^{(i)}_{\mathbf{d}(k),\mathcal{A}}\oplus W^{(k)}_{\mathbf{d}(k),\mathcal{A}}\oplus W^{(k)}_{\mathbf{d}(k),\mathcal{A}} \end{aligned}$$

for all $i\in\mathcal{K}(\mathbf{d}(k))$. Instead, if $k\neq u_{\mathbf{d}(k)}$, then he first computes

$$\begin{aligned} W^{(i)}_{\mathbf{d}(k),\mathcal{A}} &= W^{(k)}_{\mathbf{d}(k),\mathcal{A}}\oplus Y^{(k)}_{\mathbf{d}(k),\mathcal{A}} \\ &= W^{(k)}_{\mathbf{d}(k),\mathcal{A}}\oplus W^{(k)}_{\mathbf{d}(k),\mathcal{A}}\oplus W^{(u_{\mathbf{d}(k)})}_{\mathbf{d}(k),\mathcal{A}} \end{aligned}$$

for $i=u_{\mathbf{d}(k)}$, and for all 2-tuples $(i,\mathcal{A})$ such that $i\neq k$ and $(i,\mathbf{d}(k),\mathcal{A})\in\mathcal{R}_{II}(q,\mathbf{d})$, and then

$$W^{(i)}_{\mathbf{d}(k),\mathcal{A}} = W^{(u_{\mathbf{d}(k)})}_{\mathbf{d}(k),\mathcal{A}}\oplus Y^{(i)}_{\mathbf{d}(k),\mathcal{A}}$$

for all $i\in\mathcal{K}(\mathbf{d}(k))\backslash u_{\mathbf{d}(k)}$.

Next, we count the number of broadcasted subfile Type II required. Observe that, there is a broadcasted subfile Type II, $Y^{(i)}_{f,\mathcal{A}}$, for each subfile Type II except for $i=u_f$. The total number of subfiles Type II is

$$\sum_{i\in\mathcal{K}}\sum_{\mathcal{A}:\mathbf{d}(i)\in\mathcal{A}}\sum_{f\in\mathcal{A}}1 = K\binom{N_e(\mathbf{d})-1}{q-1}q \tag{13}$$

here, although not explicitly written for brevity, we require $|\mathcal{A}|=q$ and $\mathcal{A}\subseteq\mathcal{N}_e(\mathbf{d})$. The result in (13) follow since, from right to left, first, for any set $\mathcal{A}$ with $|\mathcal{A}|=q$, we have $\sum_{f\in\mathcal{A}}1=q$, second, the number of sets $\mathcal{A}$ satisfying $\mathcal{A}\subseteq\mathcal{N}_e(\mathbf{d})$ and $|\mathcal{A}|=q$ that include a particular file $\mathbf{d}(i)$ is $\sum_{\mathcal{A}:\mathbf{d}(i)\in\mathcal{A}}1=\binom{N_e(\mathbf{d})-1}{q-1}$, and, third, there are K users. Similarly, we can compute the total number of subfiles Type II with no broadcasted subfile associated to, as

$$\sum_{u\in\mathcal{U}}\sum_{\mathcal{A}:\mathbf{d}(u)\in\mathcal{A}}\sum_{f=\mathbf{d}(u)}1 = N_e(\mathbf{d})\binom{N_e(\mathbf{d})-1}{q-1} \tag{14}$$

where the set $\mathcal{U}$ contains all user leaders and thus $|\mathcal{U}|=N_e(\mathbf{d})$. Finally, subtracting (14) to (13), we have that the total number of broadcasted subfiles Type II is

$$T_{II} = (Kq-N_e(\mathbf{d}))\binom{N_e(\mathbf{d})-1}{q-1}.$$

<u>Coded cached subfiles Type III</u>: Finally, we consider the delivery of the requested subfiles Type III, $W^{(i)}_{f,\mathcal{A}}$ for all $(i,f,\mathcal{A})\in\mathcal{R}_{III}(q,\mathbf{d})$. Given that for these subfiles $\mathbf{d}(i)\notin\mathcal{A}$, we can rewrite subfiles Type III, equivalently, as $W^{(i)}_{f,\mathcal{B}\backslash\mathbf{d}(i)}$ for all $(i,f,\mathcal{B})\in\mathcal{R}'_{III}(q,\mathbf{d})$ with $\mathcal{R}'_{III}(q,\mathbf{d})$ given in (18). We show this equivalence in (15)-(17). Equality (16) follows since $\mathcal{A}=\mathcal{B}\backslash\mathbf{d}(i)$, and $\mathbf{d}(i)\in\mathcal{B}$ imply $\mathbf{d}(i)\notin\mathcal{A}$ and $\mathcal{B}=\mathcal{A}\cup\mathbf{d}(i)$, and vice-versa.

First, for each file $f\in\mathcal{B}$, we select a *user leader* $u_f\in\mathcal{K}(f)$. Next, for each set $\mathcal{B}\subseteq\mathcal{N}_e(\mathbf{d})$ with $|\mathcal{B}|=q+1$, we define an arbitrary one to one mapping function $g_{\mathcal{B}}(f)$ which for each file index $f\in\mathcal{B}$ returns a file index $g_{\mathcal{B}}(f)\in\mathcal{B}\backslash f$ satisfying $g_{\mathcal{B}}(f_1)\neq g_{\mathcal{B}}(f_2)$ if $f_1\neq f_2$. e.g. if $\mathcal{B}=\{0,1,...,q\}$, then we can use $g_{\mathcal{B}}(f)=(f-1)\mod(q+1)$. Then, the server first broadcasts

$$Y^{(i)}_{f,\mathcal{B}} = W^{(u_{g_{\mathcal{B}}(f)})}_{f,\mathcal{B}\backslash g_{\mathcal{B}}(f)}\oplus W^{(i)}_{f,\mathcal{B}\backslash\mathbf{d}(i)} \tag{19}$$

for each $(i,f,\mathcal{B})\in\mathcal{R}_{III}(q,\mathbf{d})$ with $i\neq u_{g_{\mathcal{B}}(f)}$, and then

$$Y_{\mathcal{B}} = \bigoplus_{f\in\mathcal{B}} W^{(u_{g_{\mathcal{B}}(f)})}_{f,\mathcal{B}\backslash g_{\mathcal{B}}(f)} \tag{20}$$

for each set $\mathcal{B}\subseteq\mathcal{N}_e(\mathbf{d})$ with $|\mathcal{B}|=q+1$. Although not broadcasted, let us set $Y^{(i)}_{f,\mathcal{B}}=\mathbf{0}$ for $i=u_{g_{\mathcal{B}}(f)}$.

The rationale for this broadcasting strategy is the following. Recall that coded cached subfiles Type III XOR together subfiles of files different from the requested subfile. Given a set $\mathcal{B}$, a user U_k, requesting file $\mathbf{d}(k)\in\mathcal{B}$, XORs to $Z^{(k)}_{\mathcal{B}\backslash\mathbf{d}(k)}$ all the broadcasted subfiles $Y^{(k)}_{f,\mathcal{B}}$ for all files $f\in\mathcal{B}\backslash\mathbf{d}(k)$ in order to replace $W^{(k)}_{f,\mathcal{B}\backslash\mathbf{d}(k)}$ by $W^{(u_{g_{\mathcal{B}}(f)})}_{f,\mathcal{B}\backslash g_{\mathcal{B}}(f)}$ and thus, transform the coded cached subfile $Z^{(k)}_{\mathcal{B}\backslash\mathbf{d}(k)}=\bigoplus_{f\in\mathcal{B}\backslash\mathbf{d}(k)}W^{(k)}_{f,\mathcal{B}\backslash\mathbf{d}(k)}$ into $C^{(k)}_{\mathcal{B}\backslash\mathbf{d}(k)}=\bigoplus_{f\in\mathcal{B}\backslash\mathbf{d}(k)}W^{(u_{g_{\mathcal{B}}(f)})}_{f,\mathcal{B}\backslash g_{\mathcal{B}}(f)}$. It then follows that by XORing $Y_{\mathcal{B}}$ to $C^{(k)}_{\mathcal{B}\backslash\mathbf{d}(k)}$, user U_k can obtain $W^{(u_{g_{\mathcal{B}}(\mathbf{d}(k))})}_{\mathbf{d}(k),\mathcal{B}\backslash g_{\mathcal{B}}(\mathbf{d}(k))}$, which XORed to $Y^{(i)}_{\mathbf{d}(k),\mathcal{B}}$ for all $i\neq u_{g_{\mathcal{B}}(\mathbf{d}(k))}$ returns the rest of requested subfiles type III.

Next we detail the decoding operations at user U_k to obtain the requested subfiles Type III, $W^{(i)}_{\mathbf{d}(k),\mathcal{B}\backslash\mathbf{d}(i)}$ for all $(i,\mathcal{B})$ such that $(i,\mathbf{d}(k),\mathcal{B})\in\mathcal{R}'_{III}(q,\mathbf{d})$. The decoding process begins by computing $C^{(k)}_{\mathbf{d}(k),\mathcal{B}}=Z^{(k)}_{\mathcal{B}\backslash\mathbf{d}(k)}\oplus\bigoplus_{f\in\mathcal{B}\backslash\mathbf{d}(k)}Y^{(k)}_{f,\mathcal{B}}$ for all $\mathcal{B}\subseteq\mathcal{N}_e(\mathbf{d})$ with $|\mathcal{B}|=q+1$, and $\mathbf{d}(k)\in\mathcal{B}$, and then

$$W^{(i)}_{\mathbf{d}(k),\mathcal{B}\backslash\mathbf{d}(i)} = \begin{cases} C^{(k)}_{\mathbf{d}(k),\mathcal{B}}\oplus Y_{\mathcal{B}} & \text{if } i=u_{g_{\mathcal{B}}(\mathbf{d}(k))} \\ C^{(k)}_{\mathbf{d}(k),\mathcal{B}}\oplus Y_{\mathcal{B}}\oplus Y^{(i)}_{\mathbf{d}(k),\mathcal{B}} & \text{if } i\neq u_{g_{\mathcal{B}}(\mathbf{d}(k))}, \end{cases} \tag{21}$$

for all $(i,\mathcal{B})$ such that $(i,\mathbf{d}(k),\mathcal{B})\in\mathcal{R}'_{III}(q,\mathbf{d})$.

$$\begin{aligned}\mathcal{R}_{III}(q,\mathbf{d}) &= \{(i,f,\mathcal{A})\in\mathcal{R}(q,\mathbf{d}):\ \mathcal{A}\subseteq\mathcal{N}_e(\mathbf{d}),\mathbf{d}(i)\notin\mathcal{A}\} &(15)\\ &= \{(i,f,\mathcal{A}):|\mathcal{A}|=q,\mathcal{A}\subseteq\mathcal{N}_e(\mathbf{d}),\mathbf{d}(i)\notin\mathcal{A},i\in\mathcal{K},f\in\mathcal{A}\}\\ &= \{(i,f,\mathcal{A}):|\mathcal{A}\cup\mathbf{d}(i)|=q+1,\mathcal{A}\cup\mathbf{d}(i)\subseteq\mathcal{N}_e(\mathbf{d}),\mathbf{d}(i)\notin\mathcal{A},i\in\mathcal{K},f\in\mathcal{A}\}\\ &= \{(i,f,\mathcal{A}):\mathcal{B}=\mathcal{A}\cup\mathbf{d}(i),|\mathcal{B}|=q+1,\mathcal{B}\subseteq\mathcal{N}_e(\mathbf{d}),i\in\mathcal{K},\mathbf{d}(i)\notin\mathcal{A},f\in\mathcal{B}\backslash\mathbf{d}(i)\}\\ &= \{(i,f,\mathcal{A}):\mathcal{A}=\mathcal{B}\backslash\mathbf{d}(i),\mathcal{B}\subseteq\mathcal{N}_e(\mathbf{d}),|\mathcal{B}|=q+1,i\in\mathcal{K},\mathbf{d}(i)\in\mathcal{B},f\in\mathcal{B}\backslash\mathbf{d}(i)\} &(16)\\ &= \{(i,f,\mathcal{A}):\mathcal{A}=\mathcal{B}\backslash\mathbf{d}(i),\mathcal{B}\subseteq\mathcal{N}_e(\mathbf{d}),|\mathcal{B}|=q+1,i\in\mathcal{K},\mathbf{d}(i)\in\mathcal{B}\backslash f,f\in\mathcal{B}\}\\ &= \left\{(i,f,\mathcal{A}):\mathcal{A}=\mathcal{B}\backslash\mathbf{d}(i),(i,f,\mathcal{B})\in\mathcal{R}'_{III}(q,\mathbf{d})\right\}. &(17)\end{aligned}$$

$$\mathcal{R}'_{III}(q,\mathbf{d}) = \{(i,f,\mathcal{B}):\mathcal{B}\subseteq\mathcal{N}_e(\mathbf{d}),|\mathcal{B}|=q+1,i\in\mathcal{K},\mathbf{d}(i)\in\mathcal{B}\backslash f,f\in\mathcal{B}\} \quad (18)$$

To show this result, observe that $C_{\mathbf{d}(k),\mathcal{B}}$

$$\begin{aligned}&= Z^{(k)}_{\mathcal{B}\backslash\mathbf{d}(k)}\oplus\bigoplus_{f\in\mathcal{B}\backslash\mathbf{d}(k)}Y^{(k)}_{f,\mathcal{B}}\\ &= \bigoplus_{f\in\mathcal{B}\backslash\mathbf{d}(k)}W^{(k)}_{f,\mathcal{B}\backslash\mathbf{d}(k)}\oplus\bigoplus_{f\in\mathcal{B}\backslash\mathbf{d}(k)}W^{(u_{g_\mathcal{B}(f)})}_{f,\mathcal{B}\backslash g_\mathcal{B}(f)}\oplus W^{(k)}_{f,\mathcal{B}\backslash\mathbf{d}(k)} &(22)\\ &= \bigoplus_{f\in\mathcal{B}\backslash\mathbf{d}(k)}W^{(u_{g_\mathcal{B}(f)})}_{f,\mathcal{B}\backslash g_\mathcal{B}(f)} &(23)\end{aligned}$$

and, that XORing $C_{\mathbf{d}(k),\mathcal{B}}$ and $Y_\mathcal{B}$, we can obtain

$$W^{(i)}_{\mathbf{d}(k),\mathcal{B}\backslash\mathbf{d}(i)} = C_{\mathbf{d}(k),\mathcal{B}}\oplus Y_\mathcal{B} = W^{(u_{g_\mathcal{B}(\mathbf{d}(k))})}_{\mathbf{d}(k),\mathcal{B}\backslash g_\mathcal{B}(\mathbf{d}(k))}$$

for all $(i,\mathcal{B})$ such that $i=u_{g_\mathcal{B}(\mathbf{d}(k))}$ and $(i,\mathbf{d}(k),\mathcal{B})\in\mathcal{R}'_{III}(q,\mathbf{d})$. Then, he can obtain the remaining subfiles, as

$$W^{(i)}_{\mathbf{d}(k),\mathcal{B}\backslash\mathbf{d}(i)} = W^{(u_{g_\mathcal{B}(\mathbf{d}(k))})}_{\mathbf{d}(k),\mathcal{B}\backslash g_\mathcal{B}(\mathbf{d}(k))}\oplus Y^{(i)}_{\mathbf{d}(k),\mathcal{B}}$$

for all $(i,\mathcal{B})$ with $i\neq u_{g_\mathcal{B}(\mathbf{d}(k))}$ and $(i,\mathbf{d}(k),\mathcal{B})\in\mathcal{R}'_{III}(q,\mathbf{d})$.

Next, we compute the total number of broadcasted subfiles Type III. For brevity, although not explicitly written, the summations over $\mathcal{B}$ require $\mathcal{B}\subseteq\mathcal{N}_e(\mathbf{d})$ and $|\mathcal{B}|=q+1$. Observe that there is one broadcasted subfile $Y_\mathcal{B}$ for each set $\mathcal{B}\subseteq\mathcal{N}_e(\mathbf{d})$ with $|\mathcal{B}|=q+1$, and one broadcasted subfile $Y^{(i)}_{f,\mathcal{B}}$ for all $(i,f,\mathcal{B})\in\mathcal{R}_{III}(q,\mathbf{d})$ with $i\neq u_{g_\mathcal{B}(f)}$, or more explicitly for all $(i,f,\mathcal{B})\in\left\{\mathcal{B}\subseteq\mathcal{N}_e(\mathbf{d}),|\mathcal{B}|=q+1,i\in\mathcal{K}\backslash u_{g_\mathcal{B}(f)},\mathbf{d}(i)\in\mathcal{B}\backslash f,f\in\mathcal{B}\right\}$. Thus, the total number of broadcasted subfiles Type III is

$$\begin{aligned}T_{III} &= \sum_\mathcal{B}\left(\sum_{f\in\mathcal{B}}\left(\sum_{d\in\mathcal{B}\backslash f}\sum_{i\in\mathcal{K}(d)\backslash u_{g_\mathcal{B}(f)}}1\right)+1\right)\\ &= \sum_\mathcal{B}\left(\sum_{f\in\mathcal{B}}\left(\sum_{d\in\mathcal{B}\backslash f}\mathcal{K}(d)-1\right)+1\right) &(24)\\ &= \sum_\mathcal{B}\sum_{f\in\mathcal{B}}\sum_{d\in\mathcal{B}\backslash f}\mathcal{K}(d)-\sum_\mathcal{B}\sum_{f\in\mathcal{B}}1+\sum_\mathcal{B}1\\ &= \sum_\mathcal{B}\sum_{f\in\mathcal{B}}\sum_{d\in\mathcal{B}}\mathcal{K}(d)-\sum_\mathcal{B}\sum_{f\in\mathcal{B}}(\mathcal{K}(g)+1)+\sum_\mathcal{B}1\\ &= (qK-N_e(\mathbf{d}))\binom{N_e(\mathbf{d})-1}{q}+\binom{N_e(\mathbf{d})}{q+1} &(25)\\ &= \left(qK-\left(1-\frac{1}{q+1}\right)N_e(\mathbf{d})\right)\binom{N_e(\mathbf{d})-1}{q}\end{aligned}$$

where (24) follow since for each f, $u_{g_\mathcal{B}(f)}$ is only found in $\mathcal{K}(d)$ for $d=g_\mathcal{B}(f)$, and (25) follows since, the number of sets $\mathcal{B}$ is

$$\sum_\mathcal{B}1=\binom{N_e(\mathbf{d})}{q+1},$$

$$\begin{aligned}\sum_\mathcal{B}\sum_{f\in\mathcal{B}}(\mathcal{K}(f)+1) &= \sum_{f\in\mathcal{N}_e(\mathbf{d})}(\mathcal{K}(f)+1)\sum_{\mathcal{B}:f\in\mathcal{B},}1\\ &= (K+N_e(\mathbf{d}))\binom{N_e(\mathbf{d})-1}{q},\end{aligned}$$

$$\begin{aligned}\sum_\mathcal{B}\sum_{f\in\mathcal{B}}\sum_{d\in\mathcal{B}}\mathcal{K}(d) &= (q+1)\sum_\mathcal{B}\sum_{d\in\mathcal{B}}\mathcal{K}(d)\\ &= (q+1)\sum_{d\in\mathcal{N}_e(\mathbf{d})}\mathcal{K}(d)\sum_{\mathcal{B}:d\in\mathcal{B}}1\\ &= (q+1)K\binom{N_e(\mathbf{d})-1}{q}\end{aligned}$$

Finally, (25) follow since $\binom{n}{k}=\frac{n}{k}\binom{n-1}{k-1}$.

Finally, adding together the three broadcasted subfiles types, we obtain

$$T=KN_e(\mathbf{d})\binom{N-1}{q-1}-N_e(\mathbf{d})\left(\frac{N_e(\mathbf{d})+1}{q+1}\right)\binom{N_e(\mathbf{d})-1}{q-1} \quad (26)$$

which leads to the rate (6) stated in Theorem 1.

V. Example

Consider a caching system with $N=3$ files, $K=6$ users and a caching capacity of MF bits with $M=\frac{1}{4}$, which corresponds to $q=2$. For this particular case, the best known coded caching scheme obtains the rate-memory pair $\left(\frac{28}{12}+\frac{1}{36},\frac{1}{4}\right)$ by memory sharing between the rate-memory pair $(R^*_{\text{MDS}},M_{\text{MDS}})=(R^*_{\text{CFL}},M_{\text{CFL}})=\left(\frac{5}{2},\frac{1}{6}\right)$ from (3) with $t=1$, and the rate-memory pair $(R^*_{\text{MDS}},M_{\text{MDS}})=\left(2,\frac{7}{15}\right)$ from (3) with $t=2$. Here, we show that the rate-memory pair $\left(\frac{28}{12},\frac{1}{4}\right)$ is achievable. The best known lower bounds [20, Theorem 1], and [23, Theorem 1] and [22, Remark 6] all obtain the same rate-memory pair $\left(\frac{27}{12},\frac{1}{4}\right)$.

For the running example, there are a total of $NK\binom{N-1}{q-1}=36$ subfiles $W^{(i)}_{f,\mathcal{A}}$ with $(i,f,\mathcal{A})\in\mathcal{T}(2)$ as specified in Table III. For the prefetching, as shown in Table I, if $q=2$ there are $\binom{N}{q}=\binom{3}{2}=3$ coded subfiles $Z^{(j)}_\mathcal{A}$ cached at each user $j\in\{1,2,...,6\}$. Recall that all subfiles are coded cached at one and only one user. Given the above prefetching scheme, we illustrate our proposed delivery strategy for a representative demand scenario, where users U_1 and U_4 request file W_1, users U_2 and U_5 request file W_2, and users U_3 and U_6 request file W_3. This corresponds to the demand vector $\mathbf{d}=[1,2,3,1,2,3]$. As we know from Corollary 1.1, since all files are requested, this is a worst case demand.

$(i,f,\mathcal{A}) \in \mathcal{T}(2) = \mathcal{R}(2,\mathbf{d})$					
i	f	$\mathcal{A}$	i	f	$\mathcal{A}$
1	1	$\{1,2\}$	4	1	$\{1,2\}$
		$\{1,3\}$			$\{1,3\}$
	2	$\{1,2\}$		2	$\{1,2\}$
		{2,3}			$\{2,3\}$
	3	$\{1,3\}$		3	$\{1,3\}$
		{2,3}			$\{2,3\}$
2	1	$\{1,2\}$	5	1	$\{1,2\}$
		$\{1,3\}$			$\{1,3\}$
	2	$\{1,2\}$		2	$\{1,2\}$
		$\{2,3\}$			$\{2,3\}$
	3	$\{1,3\}$		3	$\{1,3\}$
		$\{2,3\}$			$\{2,3\}$
3	1	$\{1,2\}$	6	1	$\{1,2\}$
		$\{1,3\}$			$\{1,3\}$
	2	$\{1,2\}$		2	$\{1,2\}$
		$\{2,3\}$			$\{2,3\}$
	3	$\{1,3\}$		3	$\{1,3\}$
		$\{2,3\}$			$\{2,3\}$

TABLE III: 3-tuples requested Type II $\mathcal{R}_{II}(2,\mathbf{d})$ (white), and Type III $\mathcal{R}_{III}(2,\mathbf{d})$ (gray).

$(i,f,\mathcal{A}) \in \mathcal{R}_{II}(2,\mathbf{d}) : i \neq u_f$			
i	f	$\mathcal{A}$	$Y^{(i)}_{f,\mathcal{A}}$
1	2	$\{1,2\}$	$W^{(1)}_{2,\{1,2\}}$
	3	$\{1,3\}$	$W^{(1)}_{3,\{1,3\}}$
2	1	$\{1,3\}$	$W^{(2)}_{1,\{1,3\}}$
	3	$\{2,3\}$	$W^{(2)}_{3,\{2,3\}}$
3	1	$\{1,3\}$	$W^{(3)}_{1,\{1,3\}}$
	2	$\{2,3\}$	$W^{(3)}_{2,\{2,3\}}$
4	1	$\{1,2\}$	$W^{(4)}_{1,\{1,2\}} \oplus W^{(1)}_{1,\{1,2\}}$
		{1,3}	$W^{(4)}_{1,\{1,3\}} \oplus W^{(1)}_{1,\{1,3\}}$
	2	$\{1,2\}$	$W^{(4)}_{2,\{1,2\}}$
	3	$\{1,3\}$	$W^{(4)}_{3,\{1,3\}}$
5	1	$\{1,2\}$	$W^{(5)}_{1,\{1,2\}}$
	2	$\{1,2\}$	$W^{(5)}_{2,\{1,2\}} \oplus W^{(2)}_{2,\{1,2\}}$
		$\{2,3\}$	$W^{(5)}_{2,\{2,3\}} \oplus W^{(2)}_{2,\{2,3\}}$
	3	$\{2,3\}$	$W^{(5)}_{3,\{2,3\}}$
6	1	$\{1,3\}$	$W^{(6)}_{1,\{1,3\}}$
	2	$\{2,3\}$	$W^{(6)}_{2,\{2,3\}}$
	3	$\{1,3\}$	$W^{(6)}_{3,\{1,3\}} \oplus W^{(3)}_{3,\{1,3\}}$
		$\{2,3\}$	$W^{(6)}_{3,\{2,3\}} \oplus W^{(3)}_{3,\{2,3\}}$

TABLE IV: Broadcasted subfiles Type II.

First, let us classify subfiles into the three subfile types. For the particular demand considered, given that all subfiles are requested, we have $\mathcal{T}(2) = \mathcal{R}(2,\mathbf{d})$, and thus, there are no subfiles Type I, i.e. $\mathcal{R}_I(2,\mathbf{d}) = \emptyset$. The gray cells in Table III indicate the subfiles Type III, and the white cells the subfiles Type II.

For the delivery of subfiles Type II, we first select one user leader u_f for each requested file $f \in \{1,2,3\}$, i.e. $u_1 = 1, u_2 = 2, u_3 = 3$. Then, the server broadcasts the subfiles Type II according to (9) as specified in Table IV. There are a total of 18 broadcasted subfiles Type II.

For the delivery of subfiles Type III, given that the only one set $\mathcal{B}$ satisfying $\mathcal{B} \subseteq \mathcal{N}_e(\mathbf{d})$ with $|\mathcal{B}| = q+1$ is $\{1,2,3\}$,we

$\mathcal{B} \subseteq \mathcal{N}_e(\mathbf{d})$ with $\lvert\mathcal{B}\rvert = q+1$.	
$\mathcal{B}$	$Y_{\mathcal{B}}$
{1,2,3}	$W^{(3)}_{1,\{1,2\}} \oplus W^{(1)}_{2,\{2,3\}} \oplus W^{(2)}_{3,\{1,3\}}$

$(i,f,\mathcal{B}) \in \mathcal{R}_{III}(q,\mathbf{d})$ with $i \neq u_{g_{\mathcal{B}}(f)}$					
i	f	$g_{\mathcal{B}}(f)$	$u_{g_{\mathcal{B}}(f)}$	$\mathbf{d}(i)$	$Y^{(i)}_{f,\mathcal{B}}$
1	3	2	2	1	$W^{(2)}_{3,\{1,3\}} \oplus W^{(1)}_{3,\{2,3\}}$
2	1	3	3	2	$W^{(3)}_{1,\{1,2\}} \oplus W^{(2)}_{1,\{1,3\}}$
3	2	1	1	3	$W^{(1)}_{2,\{2,3\}} \oplus W^{(3)}_{2,\{1,2\}}$
4	2	1	1	1	$W^{(1)}_{2,\{2,3\}} \oplus W^{(4)}_{2,\{2,3\}}$
	3	2	2	1	$W^{(2)}_{3,\{1,3\}} \oplus W^{(4)}_{3,\{2,3\}}$
5	1	3	3	2	$W^{(3)}_{1,\{1,2\}} \oplus W^{(5)}_{1,\{1,3\}}$
	3	2	2	2	$W^{(2)}_{3,\{1,3\}} \oplus W^{(3)}_{3,\{1,3\}}$
6	1	3	3	3	$W^{(2)}_{1,\{1,3\}} \oplus W^{(6)}_{1,\{1,2\}}$
	2	1	1	3	$W^{(1)}_{2,\{2,3\}} \oplus W^{(6)}_{2,\{1,2\}}$

TABLE V: Broadcasted subfiles Type III.

		$\mathcal{K}(\mathbf{d}(k))\backslash k = \{4\}$, $\bar{\mathcal{K}}(\mathbf{d}(k)) = \{2,3,5,6\}$	
		$k = 1 = u_{\mathbf{d}(k)}$,$u_{g_{\mathcal{B}}}(\mathbf{d}(k)) = 3$	
W_1	i	**Decoding operations**	
$W^{(1)}_{1,\{1,2\}}$	1	$Z^{(1)}_{\{1,2\}} \oplus Y^{(1)}_{2,\{1,2\}}$	$i = k$
$W^{(1)}_{1,\{1,3\}}$		$Z^{(1)}_{\{1,3\}} \oplus Y^{(1)}_{3,\{1,3\}}$	$i = k$
$W^{(2)}_{1,\{1,2\}}$	2	$Y^{(2)}_{1,\{1,2\}}$	$i \in \bar{\mathcal{K}}(\mathbf{d}(k))$
$W^{(2)}_{1,\{1,3\}}$		$W^{(3)}_{1,\{1,2\}} \oplus Y^{(2)}_{1,\{1,2,3\}}$	$i \neq u_{g_{\mathcal{B}}}(\mathbf{d}(k))$
$W^{(3)}_{1,\{1,2\}}$	3	$Z^{(1)}_{\{2,3\}} \oplus Y^{(1)}_{2,\{1,2,3\}}$ $\oplus Y^{(1)}_{3,\{1,2,3\}} \oplus Y_{\{1,2,3\}}$	$i = u_{g_{\mathcal{B}}}(\mathbf{d}(k))$
$W^{(3)}_{1,\{1,3\}}$		$Y^{(3)}_{1,\{1,3\}}$	$i \in \bar{\mathcal{K}}(\mathbf{d}(k))$
$W^{(4)}_{1,\{1,2\}}$	4	$Y^{(4)}_{1,\{1,2\}} \oplus W^{(1)}_{1,\{1,2\}}$	$i \in \mathcal{K}(\mathbf{d}(k))\backslash k$
$W^{(4)}_{1,\{1,3\}}$		$Y^{(4)}_{1,\{1,3\}} \oplus W^{(1)}_{1,\{1,3\}}$	$i \in \mathcal{K}(\mathbf{d}(k))\backslash k$
$W^{(5)}_{1,\{1,2\}}$	5	$Y^{(5)}_{1,\{1,2\}}$	$i \in \bar{\mathcal{K}}(\mathbf{d}(k))$
$W^{(5)}_{1,\{1,3\}}$		$W^{(3)}_{1,\{1,2\}} \oplus Y^{(5)}_{1,\{1,2,3\}}$	$i \neq u_{g_{\mathcal{B}}}(\mathbf{d}(k))$
$W^{(6)}_{1,\{1,2\}}$	6	$W^{(3)}_{1,\{1,2\}} \oplus Y^{(6)}_{1,\{1,2,3\}}$	$i \neq u_{g_{\mathcal{B}}}(\mathbf{d}(k))$
$W^{(6)}_{1,\{1,3\}}$		$Y^{(6)}_{1,\{1,3\}}$	$i \in \bar{\mathcal{K}}(\mathbf{d}(k))$

TABLE VI: Decoding of subfiles Type II (white) and Type III (gray) at user U_1.

define the function

$$g_{\{1,2,3\}}(f) = \begin{cases} 3 & \text{if } f = 1 \\ 1 & \text{if } f = 2 \\ 2 & \text{if } f = 3 \end{cases}$$

Then, the server broadcasts the subfiles Type III according to (19) and (20) as specified in Table V. Observe that the total number of broadcasted subfiles is 28, (18 Type II) and (1+9 Type III).

Given the above broadcasted subfiles, the decoding process for user U_1 (leader) is summarized in Table VI and for user U_3 (not leader) in Table VII. For each subfile, we indicate the conditions on the user index i, that allow us to identify which decoding operation is applied according to (10), and (11) for subfiles Type II, and according to (21) for subfiles Type III.

VI. Conclusions

In this work, we proposed a novel centralized coded caching scheme for the case where there are more users than files

		$\mathcal{K}(\mathbf{d}(k))\backslash k = \{4\}$, $\bar{\mathcal{K}}(\mathbf{d}(k)) = \{2,3,5,6\}$	
		$k = 4 \neq u_{\mathbf{d}(k)}$,$u_{\mathbf{d}(k)} = 1$, $u_{g_{\mathcal{B}}}(\mathbf{d}(k)) = 3$	
W_1	i	**Decoding operations**	
$W^{(1)}_{1,\{1,2\}}$	1	$Y^{(1)}_{1,\{1,2\}} \oplus W^{(4)}_{1,\{1,2\}}$	$i = u_{\mathbf{d}(k)}$
$W^{(1)}_{1,\{1,3\}}$		$Y^{(1)}_{1,\{1,3\}} \oplus W^{(4)}_{1,\{1,3\}}$	$i = u_{\mathbf{d}(k)}$
$W^{(2)}_{1,\{1,2\}}$	2	$Y^{(2)}_{1,\{1,2\}}$	$i \in \bar{\mathcal{K}}(\mathbf{d}(k))$
$W^{(2)}_{1,\{1,3\}}$		$W^{(3)}_{1,\{1,2\}} \oplus Y^{(2)}_{1,\{1,2,3\}}$	$i \neq u_{g_{\mathcal{B}}}(\mathbf{d}(k))$
$W^{(3)}_{1,\{1,2\}}$	3	$Z^{(4)}_{\{2,3\}} \oplus Y^{(4)}_{2,\{1,2,3\}} \oplus Y^{(4)}_{3,\{1,2,3\}} \oplus Y_{\{1,2,3\}}$	$i = u_{g_{\mathcal{B}}}(\mathbf{d}(k))$
$W^{(3)}_{1,\{1,3\}}$		$Y^{(3)}_{1,\{1,3\}}$	$i \in \bar{\mathcal{K}}(\mathbf{d}(k))$
$W^{(4)}_{1,\{1,2\}}$	4	$Z^{(4)}_{\{1,2\}} \oplus Y^{(4)}_{2,\{1,2\}}$	$i = k$
$W^{(4)}_{1,\{1,3\}}$		$Z^{(4)}_{\{1,3\}} \oplus Y^{(4)}_{3,\{1,3\}}$	$i = k$
$W^{(5)}_{1,\{1,2\}}$	5	$Y^{(5)}_{1,\{1,2\}}$	$i \in \bar{\mathcal{K}}(\mathbf{d}(k))$
$W^{(5)}_{1,\{1,3\}}$		$W^{(3)}_{1,\{1,2\}} \oplus Y^{(5)}_{1,\{1,2,3\}}$	$i \neq u_{g_{\mathcal{B}}}(\mathbf{d}(k))$
$W^{(6)}_{1,\{1,2\}}$	6	$W^{(3)}_{1,\{1,2\}} \oplus Y^{(6)}_{1,\{1,2,3\}}$	$i \neq u_{g_{\mathcal{B}}}(\mathbf{d}(k))$
$W^{(6)}_{1,\{1,3\}}$		$Y^{(6)}_{1,\{1,3\}}$	$i \in \bar{\mathcal{K}}(\mathbf{d}(k))$

TABLE VII: Decoding of subfiles Type II (white) and Type III (gray) at user U_4.

$K \geq N$ and users are equipped with small memories $M \leq \frac{N}{K}$. The scheme uses coded prefetching and outperforms previously proposed schemes for moderate-high number of users, $N \leq K \leq \frac{N^2+1}{2}$. Due to the limited region of applicability, the practical interest of this scheme might be small. However, the ideas and strategy here presented may motivate further developments in the coded caching problem. Our current and future work is in this direction, and includes the extension of the proposed coded prefetching technique to larger memories and scenarios with more users than files.

Appendix A
Proof of Corollary 1.1

To prove this result, we need to show that (26) increases monotonically as a function of $N_e(\mathbf{d})$. For the easy of notation, let us substitute $N_e(\mathbf{d})$ by x, and denote (26) as a function of $N_e(\mathbf{d})$ as $T(x)$. Then, observe that

$$\begin{aligned} T(x) &= Kx\binom{N-1}{q-1} - q\binom{x+1}{q+1} \\ &= Kx\binom{N-1}{q-1} - q\binom{x+2}{q+1}\left(1-\frac{q+1}{x+2}\right) \qquad (27) \\ &= Kx\binom{N-1}{q-1} - q\binom{x+2}{q+1} + q\frac{q+1}{x+2}\binom{x+2}{q+1} \\ &= Kx\binom{N-1}{q-1} - q\binom{x+2}{q+1} + q\binom{x+2}{q+1} \qquad (28) \\ &= T(x+1) + (x+1)\binom{x}{q-1} - K\binom{N-1}{q-1} \end{aligned}$$

where (27) follows from $\binom{n}{k} = \binom{n+1}{k}\left(1-\frac{k}{n+1}\right)$ and (28) follows by applying $\binom{n}{k} = \binom{n-1}{k-1}\frac{n}{k}$ twice. Thus, for $x = 1$ to $x = N-1$, we have

$$\begin{aligned} T(x+1) &= T(x) + K\binom{N-1}{q-1} - (x+1)\binom{x}{q-1} \\ &\geq T(x) \end{aligned}$$

where the last inequality follows since $\binom{x}{q-1}$ increases monotonically with x, and $x \leq \min\{N-1, K-1\}$. Finally, particularizing (26) to $N_e(\mathbf{d}) = \min(N, K)$, we obtain the rate-memory function in (7).

Appendix B
Proof of Corollary

If $N \geq K$ and $M \leq \frac{N}{K}$, by memory sharing between the rate memory pair $(K, 0)$ and the rate-memory pair $(\frac{K-1}{2}, \frac{N}{K})$ in [16], we obtain the rate-memory function

$$R^*_{\text{YU}}(M) = K - \frac{K}{N}\frac{K+1}{2}M. \qquad (29)$$

Particularizing (29) to the memory points $M = \frac{N}{Kq}$ for $q = 1, ..., N$, we have

$$R^*_{\text{YU}}\left(\frac{N}{Kq}\right) = K - \frac{K+1}{2q}.$$

At these memory points, if $N > K$ the scheme here proposed (7) obtains the rates

$$R^*\left(\frac{N}{Kq}\right) = K - \frac{N}{K}\frac{K+1}{q+1}\frac{\binom{K}{q}}{\binom{N}{q}}.$$

Thus, our scheme outperforms [16] $R^*_{\text{YU}}\left(\frac{N}{Kq}\right) \geq R^*\left(\frac{N}{Kq}\right)$, if $\frac{q2N}{Kq+1} \geq \frac{\binom{N}{q}}{\binom{K}{q}}$ which finally leads to the condition $\frac{2q}{q+1} \geq \prod_{i=1}^{q-1}\left(\frac{N-i}{K-i}\right)$ if $q \leq K$.

Appendix C
Proof of Corollary 1.3

To prove this results, we first observe that by isolating parameter t in the MDS rate points, $t = \frac{K}{N}(N - R^*_{\text{MDS}})$ and substituting it into the M_{MDS} memory points, we can obtain the following memory-rate function

$$M^{(\text{lb})}_{\text{MDS}}(R) = \frac{(N-R)\left((N-R)(NK-K) + N(K-N)\right)}{N^2(K-1)}. \qquad (30)$$

Due to the convexity of $M^{(\text{lb})}_{\text{MDS}}(R)$ with respect to R, we have that $M^{(\text{lb})}_{\text{MDS}}(R)$ represents a lower-bound on the MDS memory-rate region that can be obtained by the lower convex envelope of the memory-rate pairs (3).

Particularizing (30) to the rate points in (7), we can write $M^{(\text{lb})}_{\text{MDS}}$ as a function of q as

$$M^{(\text{lb})}_{\text{MDS}}(q) = \frac{N+1}{q+1}\frac{1}{K(K-1)}\left(\frac{N^2-1}{q+1} + K - N\right)$$

for $q = 1, ..., N$. Thus, a sufficient conditions for the new rate-memory pairs in (7) to improve the MDS rate-memory region is $\frac{N}{Kq} \leq M^{(\text{lb})}_{\text{MDS}}(q)$ which leads us to $K \leq \frac{N^2q+1}{q+1}$. Finally, since $\frac{N^2q+1}{q+1}$ increases monotonically with q, we have that a sufficient condition for our strategy to improve the MDS strategy if $M \in \left[0, \frac{N}{K}\right]$ and $N \leq K$, is $K \leq \frac{N^2+1}{2}$.

Appendix D
Proof of Corollary 1.4

The proof of Corollary 1.4 follows the lines of [15]. From [3], the cut set lower bound is given by

$$R_{\text{CB}}(M) = \max_{s\in\{1,\ldots,\min(N,K)\}} s - \frac{s}{\left\lfloor \frac{N}{s} \right\rfloor} M.$$

Particularized to the case $N \leq K$, we have

$$\begin{aligned} R_{\text{CB}}(M) &\geq \max\left(1 - \frac{1}{\left\lfloor \frac{N}{s} \right\rfloor} M, ..., N - NM\right) \\ &\geq N - NM \\ &= R^*(M) - \frac{N}{K}\left(\frac{N}{q} - \frac{N+1}{q+1}\right) \\ &= R^*(M) - \frac{\binom{N}{q+1}}{K\binom{N-1}{q-1}} \end{aligned}$$

where $R^*(M)$ is the rate-memory tradeoff function presented in Corollary 1.1 for $q \in \{1, ..., N\}$.

Appendix E
Proof of Corollary 1.5

The lower bound obtained in [20, Theorem 1] is given by $R_{\text{STC}}(M)$

$$= \max_{\substack{s\in\{1,...,K\} \\ l\in\{1,...,\lceil \frac{N}{s}\rceil\}}} \frac{1}{l}\left(N - sM - \frac{\mu(N-ls)^+}{s+\mu} - (N-Kl)^+\right) \tag{31}$$

where $\mu = \min\left(\frac{N-ls}{l}, K-s\right)\ \forall s, l$. Observe that for $l = 1$ and $s = N-1$, $K \geq N$, we have $\mu = \min(N-s, K-s) = N-s = 1$, and the objective in (31) reads $N-(N-1)M-\frac{1}{N}$. Thus, $R_{\text{STC}}(M) \geq N-(N-1)M-\frac{1}{N}$, which particularized to $K = N$, $M = \frac{N}{(N-1)K}$, $q = N-1$ leads us to

$$\begin{aligned} R_{\text{STC}}\left(\frac{N}{(N-1)K}\right) &\geq N - 1 - \frac{1}{N} \\ &= R^*\left(\frac{1}{N-1}\right). \end{aligned}$$

References

[1] M. R. Korupolu, C. Plaxton, and R. Rajaraman, "Placement algorithms for hierarchical cooperative caching," *Journal of Algorithms*, vol. 38, no. 1, pp. 260 – 302, 2001.

[2] A. Dan, D. Sitaram, and P. Shahabuddin, "Dynamic batching policies for an on-demand video server," *Multimedia Systems*, vol. 4, no. 3, pp. 112–121, 1996.

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

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

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

[6] J. Zhang, X. Lin, and X. Wang, "Coded caching under arbitrary popularity distributions," in *ITA workshop*, Feb 2015, pp. 98–107.

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

[8] R. Pedarsani, M. A. Maddah-Ali, and U. Niesen, "Online coded caching," *IEEE/ACM Transactions on Networking*, vol. 24, no. 2, pp. 836–845, April 2016.

[9] N. Karamchandani, U. Niesen, M. A. Maddah-Ali, and S. N. Diggavi, "Hierarchical coded caching," *IEEE Transactions on Information Theory*, vol. 62, no. 6, pp. 3212–3229, June 2016.

[10] L. Zhang, Z. Wang, M. Xiao, G. Wu, and S. Li, "Centralized caching in two-layer networks: Algorithms and limits," in *IEEE WiMob*, Oct 2016, pp. 1–5.

[11] M. M. Amiri, Q. Yang, and D. Gündüz, "Coded caching for a large number of users," in *2016 IEEE Information Theory Workshop (ITW)*, Sept 2016, pp. 171–175.

[12] S. Sahraei and M. Gastpar, "K users caching two files: An improved achievable rate," in *2016 Annual Conference on Information Science and Systems (CISS)*, March 2016, pp. 620–624.

[13] H. Ghasemi and A. Ramamoorthy, "Improved lower bounds for coded caching," in *IEEE ISIT*, June 2015, pp. 1696–1700.

[14] M. M. Amiri and D. Gündüz, "Fundamental limits of coded caching: Improved delivery rate-cache capacity tradeoff," *IEEE Transactions on Communications*, vol. 65, no. 2, pp. 806–815, Feb 2017.

[15] Z. Chen, P. Fan, and K. B. Letaief, "Fundamental limits of caching: improved bounds for users with small buffers," *IET Communications*, vol. 10, no. 17, pp. 2315–2318, 2016.

[16] Q. Yu, M. A. Maddah-Ali, and A. S. Avestimehr, "The exact rate-memory tradeoff for caching with uncoded prefetching," *IEEE Transactions on Information Theory*, vol. 64, no. 2, pp. 1281–1296, Feb 2018.

[17] C. Tian and J. Chen, "Caching and delivery via interference elimination," *IEEE Transactions on Information Theory*, vol. 64, no. 3, pp. 1548–1560, March 2018.

[18] C. Tian and K. Zhang, "From uncoded prefetching to coded prefetching in coded caching," *CoRR*, vol. abs/1704.07901, 2017. [Online]. Available: http://arxiv.org/abs/1704.07901

[19] K. Wan, D. Tuninetti, and P. Piantanida, "On caching with more users than files," in *IEEE ISIT*, July 2016, pp. 135–139.

[20] A. Sengupta, R. Tandon, and T. C. Clancy, "Improved approximation of storage-rate tradeoff for caching via new outer bounds," in *IEEE ISIT*, June 2015, pp. 1691–1695.

[21] C. Tian, "A note on the fundamental limits of coded caching," *CoRR*, vol. abs/1503.00010, 2015.

[22] Q. Yu, M. A. Maddah-Ali, and A. S. Avestimehr, "Characterizing the rate-memory tradeoff in cache networks within a factor of 2," in *IEEE ISIT*, June 2017, pp. 386–390.

[23] C. Y. Wang, S. S. Bidokhti, and M. Wigger, "Improved converses and gap-results for coded caching," pp. 2428–2432, June 2017.

Jesús Gómez-Vilardebó received his M.Sc. and Ph.D. degrees in Telecommunication Engineering from the Universitat Politècnica de Catalunya (UPC) in October 2003 and July 2009, respectively. In September 2005, he was granted by the Centre Tecnològic de Telecomunicacions de Catalunya (CTTC/CERCA) to obtain the Ph.D. on Signal Theory and Communications at the UPC. He is now with the CTTC holding a Research Associate position. His research interests are in the areas of information and communication theory, and their applications in multi-user communications, coded caching and information privacy.