# The Optimal Rate Memory Tradeoff in Multi-Access Coded Caching: Large Cache Size

Vijith Kumar K. P., Brijesh Kumar Rai, Tony Jacob

# The Optimal Rate Memory Tradeoff in Multi-Access Coded Caching: Large Cache Size

Vijith Kumar K P
Inria, France
Email: vijith-kumar.kizhakke-purakkal@inria.fr

Brijesh Kumar Rai
Ioncure Tech Private Limited
Email: brijesh.rai@gmail.com

Tony Jacob
IIT Guwahati, India
Email: tonyj@iitg.ac.in

***Abstract*—In this paper, we consider the $(N, K, L)$ multi-access caching network where $K$ users and $K$ caches are connected to a server with $N$ files, each of size $F$ bits, through a shared error-free broadcast channel. Each user has access to $L$ nearby caches, each of size $MF$ bits, in a cyclic wrap-around manner. Even after several previous attempts, the exact characterization of the optimal rate memory tradeoff is still an open problem except in the case where $L = K - 1$ and $L = 1$ with large cache $M \in [\frac{N}{L}.\frac{K-1}{K}, \frac{N}{L}]$. This paper determines the optimal rate memory tradeoff for the cache network with $L = K - 2$ and $M \in [\frac{N}{K-2}.\frac{K-1}{K}, \frac{N}{K-2}]$. This is done by proposing a new caching scheme that operates at the memory rate pair $(\frac{N}{K-2}.\frac{K-1}{K}, \frac{1}{K})$ and deriving a set of lower bounds to demonstrate the optimality of the scheme.**

***Index Terms*—Coded caching, multi-access cache network, exact rate memory tradeoff.**

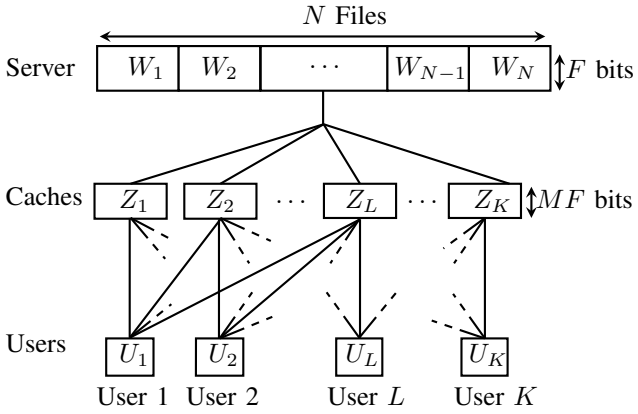


Fig. 1. The $(N, K, L)$ multi-access cache network

## I. Introduction

In their seminal work [1], Maddah-Ali and Niesen introduced the concept of coded caching for the content distribution networks. They studied the $(N, K, 1)$ canonical cache network, in which a server with $N$ files, each of size $F$ bits, is connected to the $K$ users where each user has access to a dedicated cache of size $MF$ bits, where $M \in [0, N]$. This network operates in two phases. In the first phase, called the placement phase, the server populates each cache with functions of files stored in it without any prior knowledge about users' future demands. When users reveal their demands, the server broadcasts a set of messages to help each user to obtain their requested file by using the cache contents it has access to. This phase is called the delivery phase. The main goal of the caching problem is to properly construct the placement phase so that the load experienced by the network during the delivery phase is minimal. The $(N, K)$ canonical cache network formulated in [1] was extended to study several variants in [2]–[13].

In [10], Hachem et al. considered the $(N, K, L)$ multi-access cache network, in which each user has access to $L$ nearby caches, as illustrated in Fig.1. This network has a server with $N$ files, $\{W_1, W_2, \ldots, W_N\}$, each of size $F$ bits, and is connected to $K$ users, $\{U_1, U_2, \ldots, U_K\}$, through an error-free broadcast channel. The network also has $K$ caches, $\{Z_1, Z_2, \ldots, Z_K\}$, each of size $MF$ bits where $M \in [0, N]$. Each user has access to $L$ neighboring caches in a cyclic wrap around fashion. Each cache is also accessed by $L$ users. Let $\boldsymbol{d} = [W_{d_1}, W_{d_2}, \ldots, W_{d_K}]$ represent the users' demand, with $W_{d_l}$ representing the file requested by user $U_l$ (here $d_l$ represent the file index). The server broadcasts a set of packets $X_{\boldsymbol{d}}$ of size $R_{\boldsymbol{d}}(M)F$ bits in response to the demand $\boldsymbol{d}$ to assist the user in computing their request using the cache contents it has access to. The amount $R_{\boldsymbol{d}}(M)F$ stands for the network load corresponding to the demand $\boldsymbol{d}$, and the quantity $R_{\boldsymbol{d}}(M)$ stands for the rate corresponding to the demand $\boldsymbol{d}$. If there is a caching scheme that allows each user to recover their requested file from the received packets of size at most $R(M)F$ bits for a given cache size $M$, then the memory rate pair $(M, R)$ is said to be achievable. The optimal rate memory tradeoff is defined as the minimum $R$ that allows $(M, R)$ to be achieved and it is denoted by:

$$R^*(M) = \min\{R \mid (M, R) \text{ is achievable}\} \tag{1}$$

The $(N, K, L)$ cache network was extensively studied in [11]–[21]. In [11] proposed a caching scheme and derived a set of lower bounds on the achievable rate to demonstrate that the proposed scheme is optimal among uncoded caching schemes when $L > \frac{K}{2}$. The lower bounds derived in [11] were further improved in [12]. For large cache, where $M \in [\frac{N}{L}.\frac{K-1}{K}, \frac{N}{L}]$, among all caching schemes, the schemes presented in [1] and [13] exactly characterizes the optimal rate memory tradeoff when $L = 1$ and $L = K-1$, respectively. For the other values of $L$ the exact nature of the optimal rate memory tradeoff is still an open problem. In this paper we address this issue partially by considering the cache network where each user has access to $K - 2$ nearby caches, i.e., $L = K - 2$. For this network we propose a new caching scheme which operates

TABLE I
CACHE CONTENTS PLACED IN STAGE 1 AND STAGE 2

| Cache | Stage 1 | Stage 2 |
|---|---|---|
| $Z_1$ | $A_{(1,2)}, A_{(1,3)}, A_{(1,4)}, B_{(1,2)}, B_{(1,3)}, B_{(1,4)}, C_{(1,2)}, C_{(1,3)}, C_{(1,4)}, D_{(1,2)}, D_{(1,3)}, D_{(1,4)}, E_{(1,2)}, E_{(1,3)}, E_{(1,4)}$ | $A_{(3,1)}+A_{(4,1)}+A_{(5,1)}, B_{(3,1)}+B_{(4,1)}+B_{(5,1)}, C_{(3,1)}+C_{(4,1)}+C_{(5,1)}, D_{(3,1)}+D_{(4,1)}+D_{(5,1)}, E_{(3,1)}+E_{(4,1)}+E_{(5,1)}$ |
| $Z_2$ | $A_{(2,3)}, A_{(2,4)}, A_{(2,5)}, B_{(2,3)}, B_{(2,4)}, B_{(2,5)} C_{(2,3)}, C_{(2,4)}, C_{(2,5)}, D_{(2,3)}, D_{(2,4)}, D_{(2,5)}, E_{(2,3)}, E_{(2,4)}, E_{(2,5)}$ | $A_{(4,2)}+A_{(5,2)}+A_{(1,2)}, B_{(4,2)}+B_{(5,2)}+B_{(1,2)}, C_{(4,2)}+C_{(5,2)}+C_{(1,2)}, D_{(4,2)}+D_{(5,2)}+D_{(1,2)}, E_{(4,2)}+E_{(5,2)}+E_{(1,2)}$ |
| $Z_3$ | $A_{(3,4)}, A_{(3,5)}, A_{(3,1)}, B_{(3,4)}, B_{(3,5)}, B_{(3,1)}, C_{(3,4)}, C_{(3,5)}, C_{(3,1)}, D_{(3,4)}, D_{(3,5)}, D_{(3,1)}, E_{(3,4)}, E_{(3,5)}, E_{(3,1)}$ | $A_{(5,3)}+A_{(1,3)}+A_{(2,3)}, B_{(5,3)}+B_{(1,3)}+B_{(2,3)}, C_{(5,3)}+C_{(1,3)}+C_{(2,3)}, D_{(5,3)}+D_{(1,3)}+D_{(2,3)}, E_{(5,3)}+E_{(1,3)}+E_{(2,3)}$ |
| $Z_4$ | $A_{(4,5)}, A_{(4,1)}, A_{(4,2)}, B_{(4,5)}, B_{(4,1)}, B_{(4,2)}, C_{(4,5)}, C_{(4,1)}, C_{(4,2)}, D_{(4,5)}, D_{(4,1)}, D_{(4,2)} E_{(4,5)}, E_{(4,1)}, E_{(4,2)}$ | $A_{(1,4)}+A_{(2,4)}+A_{(3,4)}, B_{(1,4)}+B_{(2,4)}+B_{(3,4)}, C_{(1,4)}+C_{(2,4)}+C_{(3,4)}, D_{(1,4)}+D_{(2,4)}+D_{(3,4)}, E_{(1,4)}+E_{(2,4)}+E_{(3,4)}$ |
| $Z_5$ | $A_{(5,1)}, A_{(5,2)}, A_{(5,3)}, B_{(5,1)}, B_{(5,2)}, B(5,3), C_{(5,1)}, C_{(5,2)}, C_{(5,3)}, D_{(5,1)}, D_{(5,2)}, D_{(5,3)} E_{(5,1)}, E_{(5,2)}, E_{(5,3)}$ | $A_{(2,5)}+A_{(3,5)}+A_{(4,5)}, B_{(2,5)}+B_{(3,5)}+B_{(4,5)}, C_{(2,5)}+C_{(3,5)}+C_{(4,5)}, D_{(2,5)}+D_{(3,5)}+D_{(4,5)}, E_{(2,5)}+E_{(3,5)}+E_{(4,5)}$ |

at the memory rate pair $(\frac{N}{K-2}.\frac{K-1}{K}, \frac{1}{K})$. We also demonstrate the optimality of the scheme by deriving a set of lower bounds.

The rest of this paper is structured as follows. In Section II, we present some key identities and notations that will be used throughout this paper. We consider the $(5,5,3)$ multi-access cache network in Section III and present the caching scheme. In Section IV, we generalize this scheme for the $(N,K,K-2)$ multi-access cache network to obtain an exact characterization of the optimal rate memory tradeoff when $M \in [\frac{N}{K-2}.\frac{K-1}{K-2}, \frac{N}{K-2}]$. We conclude the paper in section V.

## II. PRELIMINARY RESULTS

In this section we present some preliminary results and notations which are used repeatedly in the paper. We use $[L]$ to represent the set $\{1,2,\ldots,L\}$, $Z_{[L]}$ to represent the set $\{Z_1,Z_2,\ldots,Z_L\}$, and $W_{[L]}$ to represent the set $\{W_1,W_2,\ldots,W_L\}$. For integers $J$ and $I$,

$$< J >_I = \begin{cases} J \bmod I & \text{if } J \bmod I \neq 0 \\ I & \text{if } J \bmod I = 0 \end{cases}$$

Let $\mathbf{P}$ be a set, then $< \mathbf{P} >_I$ is defined as

$$< \mathbf{P} >_I = \{< l >_I : l \in \mathbf{P}\}$$

For integers $K, l, m$, consider the set $\mathbf{S}^K_{(l,m)}$ is defined as

$$\mathbf{S}^K_{(l,m)} = [K] \backslash < \{l, l-1, \ldots, l-m+1\} >_K$$

This set has the following property

$$\mathbf{S}^K_{(l,m+p)} = \mathbf{S}^K_{(l,m)} \backslash < \{l-m, \ldots, l-m-p+1\} >_K$$

Let $\mathcal{Z}_i$ denote the set of cache accessed by user $U_i$

$$\mathcal{Z}_i = \{Z_i, Z_{<i+1>_K}, \ldots, Z_{<i+L-1>_K}\} \tag{2}$$

For the $(N,K,L)$ cache network with cache size $M$, the memory rate pair $(M,R)$ is said to be achievable if there is a scheme with $R(M) \leq R$. For a such a scheme, we have

$$H(Z_l) \leq M \tag{3}$$
$$H(X_{\boldsymbol{d}}) \leq R \tag{4}$$
$$H(\mathcal{Z}_l, X_{\boldsymbol{d}}) = H(W_{d_l}, \mathcal{Z}_l, X_{\boldsymbol{d}}) \tag{5}$$
$$H(W_1,\ldots,W_N, \mathcal{Z}_l, X_{\boldsymbol{d}}) = H(W_1,\ldots,W_N), \tag{6}$$

where (3) follows from the fact that size of each cache is $M$, (4) follows from the fact that for any demand $\boldsymbol{d}$, the size of $X_{\boldsymbol{d}}$ is at most $R(M) \leq R$, (5) follows from the fact that the file $W_{\boldsymbol{d}_l}$ can be computed from $X_{\boldsymbol{d}}$ and $\mathcal{Z}_l$, and (6) follows from the fact that $\mathcal{Z}_l$ and $X_{\boldsymbol{d}}$ are functions of files $W_{[N]}$.

## III. THE $(5,5,3)$ MULTI-ACCESS CACHE NETWORK

Consider the $(5,5,3)$ multi-access cache network, which consists of a server with five files, $A$, $B$, $C$, $D$, and $E$, each of size $F$ bits, and five users, $U_1$, $U_2$, $U_3$, $U_4$ and $U_5$. The network has five caches, each of $\frac{4}{3}F$ bits in size: $Z_1$, $Z_2$, $Z_3$, $Z_4$, and $Z_5$. Three nearby caches are accessible to each user. User 1 has access to caches $Z_1$, $Z_2$, and $Z_3$, user 2 has access to caches $Z_2$, $Z_3$, and $Z_4$, user 3 has access to caches $Z_3$, $Z_4$, and $Z_5$, user 4 has access to caches $Z_4$, $Z_5$, and $Z_1$, and user 5 has access to caches $Z_5$, $Z_1$, and $Z_2$. Each cache is accessed by three users. During the placement phase the server splits each file into 15 non-overlapping subfiles, each of size $\frac{1}{15}F$ bits. The subfiles are:

| File | Subfiles |
|---|---|
| $A$ | $A_{(1,2)}, A_{(1,3)}, A_{(1,4)}, A_{(2,3)}, A_{(2,4)}, A_{(2,5)}, A_{(3,4)}, A_{(3,5)}, A_{(3,1)}, A_{(4,5)}, A_{(4,1)}, A_{(4,2)}, A_{(5,1)}, A_{(5,2)}, A_{(5,3)}$ |
| $B$ | $B_{(1,2)}, B_{(1,3)}, B_{(1,4)}, B_{(2,3)}, B_{(2,4)}, B_{(2,5)}, B_{(3,4)}, B_{(3,5)}, B_{(3,1)}, B_{(4,5)}, B_{(4,1)}, B_{(4,2)}, B_{(5,1)}, B_{(5,2)}, B_{(5,3)}$ |
| $C$ | $C_{(1,2)}, C_{(1,3)}, C_{(1,4)}, C_{(2,3)}, C_{(2,4)}, C_{(2,5)}, C_{(3,4)}, C_{(3,5)}, C_{(3,1)}, C_{(4,5)}, C_{(4,1)}, C_{(4,2)}, C_{(5,1)}, C_{(5,2)}, C_{(5,3)}$ |
| $D$ | $D_{(1,2)}, D_{(1,3)}, D_{(1,4)}, D_{(2,3)}, D_{(2,4)}, D_{(2,5)}, D_{(3,4)}, D_{(3,5)}, D_{(3,1)}, D_{(4,5)}, D_{(4,1)}, D_{(4,2)}, D_{(5,1)}, D_{(5,2)}, D_{(5,3)}$ |
| $E$ | $E_{(1,2)}, E_{(1,3)}, E_{(1,4)}, E_{(2,3)}, E_{(2,4)}, E_{(2,5)}, E_{(3,4)}, E_{(3,5)}, E_{(3,1)}, E_{(4,5)}, E_{(4,1)}, E_{(4,2)}, E_{(5,1)}, E_{(5,2)}, E_{(5,3)}$ |

TABLE II
FILE RECOVERY

| Cache Contents | | Received Packets | Computed Subfile |
|---|---|---|---|
| Uncoded | Coded | | |
| $P_{(2,5)},\ P_{(3,5)}$ $T_{(1,4)}, T_{(2,4)}, T_{(3,4)}$ | $Q_{(3,1)} + Q_{(4,1)} + Q_{(5,1)}$ $R_{(4,2)} + R_{(5,2)} + R_{(1,2)}$ $S_{(5,3)} + S_{(1,3)} + S_{(2,3)}$ | $P_{(2,5)} + P_{(3,5)} + P_{(4,5)} + Q_{(3,1)} + Q_{(4,1)} + Q_{(5,1)}$ $+R_{(4,2)} + R_{(5,2)} + R_{(1,2)} + S_{(5,3)} + S_{(1,3)} + S_{(2,3)}$ $+T_{(1,4)} + T_{(2,4)} + T_{(3,4)}$ | $P_{(4,5)}$ |
| $P_{(3,1)}, Q_{(1,2)},\ R_{(2,3)}$ $S_{(1,4)}, S_{(2,4)}, T_{(2,5)}, T_{(3,5)}$ | $Q_{(4,2)} + Q_{(5,2)} + Q_{(1,2)}$ $R_{(5,3)} + R_{(1,3)} + R_{(2,3)}$ | $P_{(3,1)} + P_{(4,1)} + Q_{(4,2)} + Q_{(5,2)} + R_{(5,3)} + R_{(1,3)}$ $+S_{(1,4)} + S_{(2,4)} + T_{(2,5)} + T_{(3,5)}$ | $P_{(4,1)}$ |
| $Q_{(1,3)}, Q_{(2,3)}, R_{(1,4)}, S_{(2,5)}, T_{(3,1)}$ | $Q_{(5,3)} + Q_{(1,3)} + Q_{(2,3)}$ | $P_{(4,2)} + Q_{(5,3)} + R_{(1,4)} + S_{(2,5)} + T_{(3,1)}$ | $P_{(4,2)}$ |

We have chosen the notation $W_{(i,j)}$ to represent the subfiles to simplify further explanations. There are two stages to the placement phase. The server copies 15 uncoded subfiles during the first stage of the placement phase. During the second stage of the placement phase, the server copies 5 functions of subfiles. TABLE I shows the cache contents for each user after the placement phase. It can be noted that during the placement phase the server copies 20 packets, each of size $\frac{1}{15}F$ bits, into each cache. These packets collectively occupies the space of $\frac{4}{3}F$ bits.

Consider a demand $\boldsymbol{d} = [P, Q, R, S, T]$, where $P, Q, R, S, T \in \{A, B, C, D, E\}$. In this demand, $P$ represents the file requested by user 1, $Q$ represents the file requested by user 2, $R$ represents the file requested by user 3, $S$ represents the file requested by user 4, and $T$ represents the file requested by user 5. In response to the demand $\boldsymbol{d}$ the server broadcasts a set of packets:

$$X_{\boldsymbol{d}} = \left\{ \begin{array}{c} P_{(2,5)} + P_{(3,5)} + P_{(4,5)} + Q_{(3,1)} + Q_{(4,1)} + Q_{(5,1)} \\ +R_{(4,2)} + R_{(5,2)} + R_{(1,2)} + S_{(5,3)} + S_{(1,3)} + S_{(2,3)} \\ +T_{(1,4)} + T_{(2,4)} + T_{(3,4)} \\ P_{(3,1)} + P_{(4,1)} + Q_{(4,2)} + Q_{(5,2)} + R_{(5,3)} + R_{(1,3)} \\ +S_{(1,4)} + S_{(2,4)} + T_{(2,5)} + T_{(3,5)} \\ P_{(4,2)} + Q_{(5,3)} + R_{(1,4)} + S_{(2,5)} + T_{(3,1)} \end{array} \right\}$$

It can be noted that the broadcast message consists of three packets each of size $\frac{1}{15}F$ bits. Thus, the load corresponding to the demand $\boldsymbol{d}$ is

$$3 \times \frac{1}{15}F = \frac{1}{5}F \text{ bits}$$

and the corresponding rate is $R = \frac{1}{5}$.

Consider user 1 in order to better understand how each user recovers the requested file from the received packets $X_{\boldsymbol{d}}$ using the cache contents it has access to. User 1 has access to the caches $Z_1, Z_2$, and $Z_3$. Thus, the subfiles $P_{(1,2)}$, $P_{(1,3)}$, $P_{(1,4)}$, $P_{(2,3)}$, $P_{(2,4)}$, $P_{(2,5)}, P_{(3,4)}, P_{(3,5)}$, and $P_{(3,1)}$ are accessible to user 1, but in order to compute the file $P$, it needs the subfiles $P_{(4,5)}, P_{(4,1)}, P_{(4,2)}, P_{(5,1)}, P_{(5,2)}$, and $P_{(5,3)}$. With the use of the cached packets, the user computes the subfiles $P_{(4,5)}, P_{(4,1)}$, and $P_{(4,2)}$ from the received packets, as shown in TABLE II. Once the user obtains the subfile $P_{(4,1)}$, combining this subfile with the cached packets $P_{(3,1)}$ and $P_{(3,1)} + P_{(4,1)} + P_{(5,1)}$, the user computes the subfile $P_{(5,1)}$. Similarly, by combining the subfile $P_{(4,2)}$ with the cache contents $P_{(1,2)}$ and $P_{(4,2)} + P_{(5,2)} + P_{(1,2)}$, the user computes the subfile $P_{(5,2)}$. The user computes the remaining subfile, $P_{(5,3)}$, by combining the caches subfiles $P_{(1,3)}$ and $P_{(2,3)}$ with the cached packet $P_{(5,3)} + P_{(1,3)} + P_{(2,3)}$. In a similar fashion, other users compute their requested files. We summarise as:

**Lemma 1.** *The memory rate pair* $(\frac{4}{3}, \frac{1}{5})$ *is achievable for the* $(5, 5, 3)$ *multi-access cache network.*

For large cache size, where $M \in [\frac{4}{3}, \frac{5}{3}]$, all memory rate pairs $(M, \frac{1}{5} - \frac{3}{5}(M - \frac{4}{3}))$ can be achieved by memory sharing the proposed scheme and the scheme proposed in [13] that achieves the memory rate pair $(\frac{5}{3}, 0)$, where all files are available in the caches. We have a matching lower bound using cut-set arguments:

**Lemma 2.** *For the* $(5, 5, 3)$ *multi-access cache network, the achievable memory rate pair* $(M, R)$ *must satisfy the constraint*

$$3M + 5R \geq 5$$

*Proof.* Here, we provide proof of the lemma for the purpose of completeness. Let $X_1^p$ represent a set of packets broadcast in response to a demand, where user 1 request for file $p \in \{A, B, C, D, E\}$. We have

$$\begin{aligned} 3M + 5R \geq & H(Z_1) + H(Z_2) + H(Z_3) + H(X_1^A) \\ & + H(X_1^B) + H(X_1^C) + H(X_1^D) + H(X_1^E) \\ \overset{(a)}{\geq} & H(Z_1, Z_2, Z_3, X_1^A, X_1^B, X_1^C, X_1^D, X_1^E) \\ \overset{(b)}{\geq} & H(\mathcal{Z}_1, X_1^A, X_1^B, X_1^C, X_1^D, X_1^E) \\ \overset{(c)}{=} & H(A, B, C, D, E, \mathcal{Z}_1, X_1^A, X_1^B, X_1^C, X_1^D, X_1^E) \\ \overset{(d)}{=} & H(A, B, C, D, E) = 5 \end{aligned}$$

where $(a)$ follows from sub-modularity property of entropy, $(b)$ follows from (2), $(c)$ follows from (5), and $(d)$ follows from (6). □

We summarise as:

TABLE III
FILE RECOVERY

| Constraints | | Cache Contents | | Compute Packet |
|---|---|---|---|---|
| | | Uncoded | Coded | |
| $1 \le j \le K-p-2$ | $p=0$ | $-$ | $\sum_{q \in \mathbf{S}^K_{(i+j+p,2)}} W_{d_{i+j},(q,<i+j+p-1>_K)}$ | $\sum_{q \in \mathbf{S}^K_{(i+j+p,p+2)}} W_{d_{i+j},(q,<i+j+p-1>_K)}$ |
| | $p \neq 0$ | $W_{d_{i+j},(q,<i+j+p-1>_K)}$, where $q \in <\{i+j-1,\ldots,i+j+p-2\}>_K$ | | |
| $K-p-1 \le j \le K-1$ | | $W_{d_{i+j},(q,<i+j+p-1>_K)}$, where $q \in \mathbf{S}^K_{(i+j+p,p+2)}$ | | |

**Theorem 1.** *For the $(5,5,3)$ multi-access cache network, the optimal rate memory tradeoff, when $M \in \left[\frac{4}{3}, \frac{5}{3}\right]$, is given by*

$$R = 1 - \frac{3}{5}M.$$

The above theorem is a special case of Theorem 3.

## IV. NEW CACHING SCHEME

In this section we extent the caching scheme to the $(N, K, K-2)$ cache network to achieve the memory rate pair $(\frac{N}{K-2}.\frac{K-1}{K}, \frac{1}{K})$. Then we derive a matching lower bound to demonstrate the proposed scheme is optimal.

### A. Placement phase

During the placement phase, the server split each file into $K(K-2)$ subfiles, each of size $\frac{1}{K(K-2)}F$ bits. The subfiles of file $W_n$ are:

$$W_{n,(i,j)} \text{ where } i \in [K] \text{ and } j \in \mathbf{S}^K_{(i,2)}$$

The placement phase takes place in two stages. In the first stage, the server copies the uncoded subfiles $W_{n,(i,j)}$, for all $n \in [N]$ and $j \in \mathbf{S}^K_{(i,2)}$, into cache $Z_i$. In the second stage, the server copies several functions of the subfiles into each cache resulting in the cache $Z_i$ having the contents:

| Stage | Cached packet | Constraint | Number |
|---|---|---|---|
| Stage 1 | $W_{n,(i,j)}$ | $n \in [N]$ and $j \in \mathbf{S}^K_{(i,2)}$ | $N(K-2)$ |
| stage 2 | $\sum_{l \in \mathbf{S}^K_{(i+1,2)}} W_{n,(l,i)}$ | $n \in [N]$ | $N$ |

It can be noted that the subfiles $W_{n,(i,j)}$, for all $n \in [N]$ and $j \in \mathbf{S}^K_{(i,2)}$, are available at cache $Z_i$ in uncoded form and the subfiles $W_{n,(l,i)}$, for all $n \in [N]$ and $l \in \mathbf{S}^K_{(i+1,2)}$, are available at cache $Z_i$ in coded form. The total number of packets, each of size $\frac{1}{K(K-2)}F$ bits, copied in each cache is

$$N(K-2) + N = N(K-1)$$

and these subfiles occupies the space of $\frac{N}{(K-2)}.\frac{K-1}{K}F$ bits.

### B. Delivery phase

Let the server receive a demand $\boldsymbol{d}$, where each user reveals their demand. During the delivery phase the server broadcasts a set of packets:

$$X_{\boldsymbol{d}} = \bigcup_{p=0}^{K-3} \sum_{m \in [K]} \left[ \sum_{q \in \mathbf{S}^K_{(m+p,p+2)}} W_{d_m,(q,<m+p-1>_K)} \right]$$

In response to demand $\boldsymbol{d}$, the server broadcasts $K-2$ packets, each of size $\frac{1}{K(K-2)}F$ bits. Thus, the rate corresponding to the demand $\boldsymbol{d}$ is:

$$R = (K-2) \times \frac{1}{K(K-2)} = \frac{1}{K}$$

### C. File recovery

To understand how each user computes its requested file from the received packet $X_{\boldsymbol{d}}$ with the help of the cache contents it has access to, consider user $U_k$ who requests the file $W_{d_k}$. The user has access to the set of caches

$$\mathcal{Z}_k = \{Z_u : u \in \mathbf{S}^K_{(k-1,2)}\}$$

As a result, the user has access to the subfiles $W_{d_k,(i,j)}$, where $i \in \mathbf{S}^K_{(k-1,2)}$ and $j \in \mathbf{S}^K_{(i,2)}$. The user also has access to the subfiles $W_{n,(i,j)}$, where $n \in [N]$, $i \in \mathbf{S}^K_{(k-1,2)}$ and $j \in \mathbf{S}^K_{(i,2)}$. The user needs the subfiles $W_{d_k,(r,s)}$, where $r \in <\{k-2, k-1\}>_K$ and $s \in \mathbf{S}^K_{(r,2)}$, to compute file $W_{d_k}$, and the user recovers these subfiles in two stages. In the first stage, the user computes the subfile $W_{d_k,(k-2,s)}$, where $s \in \mathbf{S}^K_{(k-2,2)}$. For $0 \le p \le K-3$ and $1 \le j \le K-1$, the user computes the packet

$$\sum_{q \in \mathbf{S}^K_{(k+j+p,p+2)}} W_{d_{k+j},(q,<k+j+p-1>_K)}, \tag{7}$$

by combining the cached packets available to it, as shown in TABLE III. Consider the received packet

$$\sum_{m \in [K]} \left[ \sum_{q \in \mathbf{S}^K_{(m+p,p+2)}} W_{d_m,(q,<m+p-1>_K)} \right] \tag{8}$$

where $0 \le p \le K-3$. Let $m =< k+j >_K$ where $0 \le j \le K-1$. By combining (7) and (8), the user computes the packet

$$\sum_{q \in \mathbf{S}^K_{(k+p,p+2)}} W_{d_i,(q,<k+p-1>_K)}. \tag{9}$$

From (9), the user computes the subfile $W_{d_k,(<k-2>_K,<k+p-1>_K)}$ with the aid of the caches subfiles $W_{d_k,(q,<k+p-1>_K)}$, where $q \in \mathbf{S}^K_{(k+p,p+3)}$. Now the user has the subfiles

$$W_{d_k,(q,r)} \tag{10}$$

where $q \in [K]\backslash\{k-1\}$ and $r \in \mathbf{S}^K_{(q,2)}$. Now the user needs the subfiles $W_{d_k,(k-1,l)}$, where $l \in \mathbf{S}^K_{(k-1,2)}$, which are available to the user in coded form

$$\sum_{t \in \mathbf{S}^K_{(l+1,2)}} W_{d_k,(t,l)} \tag{11}$$

for $l \in \mathbf{S}^K_{(k-1,2)}$. The user computes the subfiles $W_{d_k,(k-1,l)}$ by combining (10) and (11). Using all the recovered subfiles the user can reconstruct the requested file $W_{d_k}$.
These observations can be summarized as:

**Theorem 2.** *The memory rate pair* $(\frac{N}{K-2}.\frac{K-1}{K}, \frac{1}{K})$ *is achievable for the* $(N, K, K-2)$ *multi-access cache network.*

### *D. Matching lower bound*

We derive a matching lower bound in the following theorem:

**Theorem 3.** *For the* $(N, K, K-2)$ *multi-access cache network, the achievable memory rate pair* $(M, R)$ *must satisfy the constraint*

$$(K-2)M + NR \geq N$$

*Proof.* Let $X_1^p$ represent a set of packets broadcast in response a demand where user 1 request for file $W_p \in W_{[N]}$. Let $X_1^{[N]} = \{X_1^1, X_1^2, \ldots, X_1^N\}$. Now we have

$$\begin{aligned}(K-2)M + NR &\geq H(Z_{[K-2]}) + H(X_1^{[N]}) \\ &\overset{(a)}{\geq} H(Z_{[K-2]}, X_1^{[N]}) \\ &\overset{(b)}{=} H(\mathcal{Z}_1, X_1^{[N]}) \\ &\overset{(c)}{=} H(W_{[N]}, \mathcal{Z}_1, X_1^{[N]}) \\ &\overset{(d)}{=} H(W_{[N]}) = N\end{aligned}$$

where $(a)$ follows from sub-modularity property of entropy, $(b)$ follows from (2), $(c)$ follows from (5), and $(d)$ follows from (6). □

We summarise as:

**Theorem 4.** *For the* $(N, K, K-2)$ *multi-access cache network, the optimal rate memory tradeoff, when* $M \in \left[\frac{N}{K-2}.\frac{K-1}{K}, \frac{N}{K-2}\right]$, *is given by*

$$R = 1 - \frac{(K-2)}{N}M.$$

For large cache size, where $M \in [\frac{N}{K-2}.\frac{K-1}{K}, \frac{N}{K-2}]$, all memory rate pairs $(M, \frac{1}{K} - \frac{K-2}{N}(M - \frac{N}{K-2}\frac{K-1}{K}))$ can be achieved by memory sharing the proposed scheme and the scheme proposed in [13] that achieves the memory rate pair $(\frac{N}{K-2}, 0)$.

## V. Conclusions

In this paper, we considered the $(5, 5, 3)$ multi-access cache network where each user has access to 3 nearby caches. For this network, we proposed a new caching scheme that operates at the memory rate pair $(\frac{4}{3}, \frac{1}{5})$, and also derived a lower bound, demonstrating the proposed scheme's optimality. We extended this scheme and the lower bound derivation for the $(N, K, K-2)$ multi-access cache network. This proved the optimality of the proposed scheme and lead to a characterization of the exact rate memory tradeoff for the multi-access cache network for large cache size.


## Acknowledgements

This research was supported in part by the French Government through the "Plan de Relance" and "Programme d'investissements d'avenir".


## References


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

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

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

[4] 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, 2016.

[5] S. P. Shariatpanahi, S. A. Motahari, and B. H. Khalaj, "Multi-server coded caching," *IEEE Transactions on Information Theory*, vol. 62, no. 12, pp. 7253–7271, 2016.

[6] E. Parrinello and P. Elia, "Coded caching with optimized shared-cache sizes," in *2019 IEEE Information Theory Workshop (ITW)*. IEEE, 2019, pp. 1–5.

[7] A. Malik, B. Serbetci, E. Parrinello, and P. Elia, "Fundamental limits of stochastic caching networks," *arXiv preprint arXiv:2005.13847*, 2020.

[8] Ç. Yapar, K. Wan, R. F. Schaefer, and G. Caire, "On the optimality of d2d coded caching with uncoded cache placement and one-shot delivery," *IEEE Transactions on Communications*, vol. 67, no. 12, pp. 8179–8192, 2019.

[9] K. Wan, D. Tuninetti, M. Ji, G. Caire, and P. Piantanida, "Fundamental limits of decentralized data shuffling," *IEEE Transactions on Information Theory*, vol. 66, no. 6, pp. 3616–3637, 2020.

[10] J. Hachem, N. Karamchandani, and S. N. Diggavi, "Coded caching for multi-level popularity and access," *IEEE Transactions on Information Theory*, vol. 63, no. 5, pp. 3108–3141, 2017.

[11] K. S. Reddy and N. Karamchandani, "Rate-memory trade-off for multi-access coded caching with uncoded placement," *IEEE Transactions on Communications*, 2020.

[12] K. Krishnan Namboodiri and B. Sundar Rajan, "An improved lower bound for multi-access coded caching," *arXiv e-prints*, pp. arXiv–2201, 2022.

[13] S. Sasi and B. S. Rajan, "An improved multi-access coded caching with uncoded placement," *arXiv preprint arXiv:2009.05377*, 2020.

[14] E. Peter and B. S. Rajan, "Coded caching with shared caches from generalized placement delivery arrays," in *2021 IEEE 32nd Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*. IEEE, 2021, pp. 380–386.

[15] E. Parrinello, A. Unsal, and P. Elia, "Fundamental limits of caching in heterogeneous networks with uncoded prefetching," *arXiv preprint arXiv:1811.06247*, 2018.

[16] A. M. Ibrahim, A. A. Zewail, and A. Yener, "Coded caching for heterogeneous systems: An optimization perspective," *IEEE Transactions on Communications*, vol. 67, no. 8, pp. 5321–5335, 2019.

[17] M. Dutta and A. Thomas, "Decentralized coded caching for shared caches," *IEEE Communications Letters*, vol. 25, no. 5, pp. 1458–1462, 2021.

[18] F. Brunero and P. Elia, "Fundamental limits of combinatorial multi-access caching," *arXiv preprint arXiv:2110.07426*, 2021.

[19] A. Malik, B. Serbetci, E. Parrinello, and P. Elia, "Fundamental limits of stochastic shared-cache networks," *IEEE Transactions on Communications*, vol. 69, no. 7, pp. 4433–4447, 2021.

[20] P. N. Muralidhar, D. Katyal, and B. S. Rajan, "Maddah-Ali-Niesen scheme for multi-access coded caching," in *2021 IEEE Information Theory Workshop (ITW)*. IEEE, 2021, pp. 1–6.

[21] B. Serbetci, E. Parrinello, and P. Elia, "Multi-access coded caching: gains beyond cache-redundancy," in *2019 IEEE Information Theory Workshop (ITW)*. IEEE, 2019, pp. 1–5.