Cache-Aided Multi-Access Multi-User Private Information Retrieval

Kanishak Vaidya, and B Sundar Rajan
Department of Electrical Communication Engineering, IISc Bangalore, India
E-mail: {kanishakv, bsrajan}@iisc.ac.in

***Abstract*—In this paper, we consider the multi-access cache-aided multi-user Private Information Retrieval (MuPIR) problem. In this problem, N files are replicated across S servers. There are K users and C cache nodes, each capable of storing M files. Each user can access L cache nodes, and every cache node can be accessed by several users. Each user wants to retrieve one file from the servers, but users don't want the servers to know their demands. This problem is an extension of the dedicated cache-aided MuPIR problem, which itself generalizes the single-user PIR setup. In this paper, we propose an order optimal MuPIR scheme that utilizes a multi-access setup of the coded caching problem where every set of L caches is accessed by one user resulting in $K = \binom{C}{L}$. We also propose an achievable scheme for the multi-access setup with cyclic wraparound cache access.**

I. Introduction

The problem of Private Information Retrieval (PIR), first described in [1] deals with privately retrieving data from distributed servers. A user wishes to retrieve one file amongst a set of files stored across the servers. But the servers should not know the identity of the desired file. A PIR scheme that minimizes the download cost for the user is described in [2]. After that, the PIR problem is solved for various other settings [3], [4], [5].

Currently, PIR is being studied with another content delivery scenario called coded caching. As described in [6], in coded caching, there are multiple users equipped with a user cache and one server storing some files. During off-peak hours, users fill their caches and then, during peak network traffic hours, demand files from the server. The server will perform coded transmissions such that a single transmission can benefit multiple users simultaneously. After receiving the transmissions, users will be able to decode their demanded files with the help of the content stored in their cache. Recently, in [7] a cache-aided PIR strategy is described where multiple users, each having access to dedicated caches, want to privately recover files from non-colluding servers. An order-optimal strategy is described that combines the coding benefits of PIR in [2] and coded caching [6].

In this paper, we use a variation of coded caching known as multi-access coded caching in PIR. In multi-access coded caching, users don't have access to dedicated caches. Instead, there are helper cache nodes, which are accessed by the users. One helper cache can be accessed by multiple users, and users can access multiple caches. We will be using the multi-access setup described in [8] which generalizes the Maddah-Ali Niesen coded caching scheme [6].

Notations: For integers m and n, $[m : n]$ is the set of integers $\{m, m+1, \cdots n\}$. $[N]$ is same as $[1 : N]$. For a set $\mathcal{S}$ of size $|\mathcal{S}|$ and an integer $N \leq |\mathcal{S}|$, $\binom{\mathcal{S}}{N}$ denotes the set of all subsets of $\mathcal{S}$ of size N. For set $\{a_n | n \in [N]\}$ and $\mathcal{N} \subseteq [N]$, $a_{\mathcal{N}}$ denotes the set $\{a_n | n \in \mathcal{N}\}$.

We will first briefly describe the single user PIR of [2] and the multi-access coded caching setup of [8] in the following two subsections.

A. Private Information Retrieval [2]

In Private Information Retrieval (PIR) there is one user and a set of N files $\mathcal{W} = \{W_n\}_{n=1}^{N}$ replicated across S non-colluding servers. The user wants to retrieve one out of N files, say the file $W_\theta, \theta \in \{1, 2, \ldots, N\}$, but doesn't want the servers to know the identity of the file. In other words, the user wants to hide the index θ from the servers. In order to retrieve this desired file privately, the user generates S queries $\{Q_s^\theta\}_{s=1}^{S}$ and sends the query Q_s^θ to server s. After receiving their respective queries, the servers construct answers which are a function of the query they got and the files they have. The server s constructs the answer $A_s^\theta(\mathcal{W})$ and sends it to the user. After receiving the answers from all S servers, the user should be able to decode its desired file. Privacy and correctness conditions are formally stated as follows: For privacy, we need that

$$I(\theta; Q_s^\theta) = 0, \forall s \in \{1, \ldots, S\},$$

and for correctness

$$H(W_\theta | \theta, A_1^\theta(\mathcal{W}) \ldots A_S^\theta(\mathcal{W}), Q_1^\theta \ldots Q_S^\theta) = 0.$$

The rate of PIR is a parameter that describes the download cost for the user (or the transmission cost for the servers), defined as

$$R_{PIR} = \frac{\sum_{s=1}^{S}(H(A_s^\theta(\mathcal{W})))}{H(W_\theta)}.$$

A rate optimal scheme is provided in [2] with optimal rate R_{PIR}^* given by

$$R_{PIR}^*(S, N) = 1 + \frac{1}{S} + \frac{1}{S^2} + \ldots + \frac{1}{S^{N-1}}.$$

B. Multi-Access Coded caching [8]

In a multi-access coded caching scenario, a server stores N files $\{W_n\}_{n \in [N]}$ each of unit size. There are K users connected via an error-free shared link to the server. There are C helper caches, each capable of storing M units. Each user has access to L out of C helper caches, and there is only one user corresponding to every choice of L out of C caches resulting in $K = \binom{C}{L}$. Let $\mathcal{Z}_k \subset [C]$ be the indices of helper

ISBN 978-3-903176-49-2 © 2022 IFIP

caches that the user k has access to. The system operates in two phases.

Placement Phase: In this phase, all C helper cache are filled without the knowledge of future demands of the users.

Delivery Phase: In this phase, each user wishes to retrieve a file from the server. User k will choose $d_k \in [N]$ and wish to retrieve W_{d_k}. Users will convey their demands to the server, and the server will perform coded transmissions such that each user gets the demanded file using the transmissions and the cache content they have access to.

The goal in this setup is to design a placement and delivery scheme such that the transmissions from the server are minimized. A placement and delivery scheme is given in [8] for this setup. This placement and delivery scheme generalizes the well-known Maddah-Ali Niesen (MAN) scheme for dedicated cache systems, i.e., for $L = 1$ and $C = K$, the scheme in [8] becomes the MAN scheme. Every file is divided into $\binom{C}{t}$ non overlapping subfiles of equal size and the server transmits $\binom{C}{L+t}$ such subfiles where $t = \frac{CM}{N} \in \mathbb{Z}$. The rate achieved in this scheme is $\frac{\binom{C}{L+t}}{\binom{C}{t}}$.

In [9] this rate is shown to be optimal under the assumption of uncoded cache placement. We denote this rate by

$$R^*_{nPIR}(t) = \frac{\binom{C}{L+t}}{\binom{C}{t}} \tag{1}$$

where the subscript ***nPIR*** is to indicate that the rate is for the multi-access setup of [8] with *no PIR* constraints.

C. Cyclic Wraparound multi-access setup

In cyclic wraparound multi-access setup numbe of users and numbe of cache nodes are equal. User k access cache nodes indexed by $\{k, k+1, \ldots, k+L-1\}$ where addition is modulo C except $k + l = C$ if $k + l$ is multiple of C. Multi-access systems with cyclic wraparound cache access are widely studied in [11], [12], [13], [14], [15], [16].

D. Our Contributions

Multi-User Private Information Retrieval (MuPIR) problem has been studied with a coded caching setup in [7]. In MuPIR, there are multiple users, each equipped with a cache and multiple servers. Each user wants to retrieve a file from the servers but doesn't want the servers to know the user's demand.

In this paper, we develop a PIR scheme in which multiple users aided with multi-access cache nodes privately retrieve data from distributed servers. We will consider the multi-access setup of [8] but with multiple non-colluding servers with all the messages replicated across all the servers. Each server is connected to all the users via broadcast links.

The contributions and the outline of this paper follow.

- In Section II we describe the system model in detail and provide formal description of privacy and correctness constraints.
- In Section III we present our results.
  - In Theorem 1 we present an achievable rate for the multiuser PIR problem with multi-access setups of [8]. The achievable scheme is given in Section IV.
  - In Theorem 2 we show that the rate achieved in Theorem 1 is order optimal within a multiplicative factor of 2 assuming uncoded storage.
  - In Theorem 3 we stated rate for multi-access system with cyclic wraparound cache access.
- We have provided detailed comparison of per user rate [10] of our setup with dedicated cache setup of [7].
- Section IV contains the proofs of privacy and achievable rates mentioned in Theorem 1 and Theorem 3.

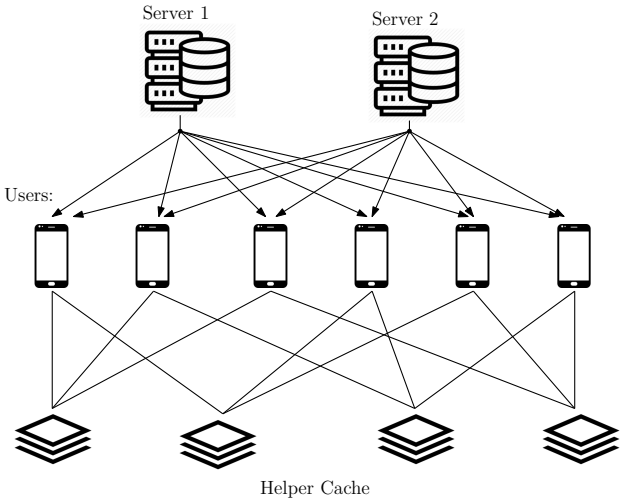

Fig. 1: Multi Access coded caching setup with two servers, four helper cache and six users. Each user have access to two helper cache

II. System Model

There are K users and N files, $\{W_n\}_{n\in[N]}$ replicated across $S \geq 2$ servers. Each file is of unit size. There are C cache nodes, each capable of storing M units. Every user is connected to $L < C$ cache nodes through an infinite capacity link. Let $\mathcal{L}_k$ be the set of indices of cache nodes that user k is connected to. In this paper we will consider two cache-user connectivities

1) Cache user connectivity of [8] as described in Section I-B. In this case, $\{\mathcal{L}_k : k \in [K]\} = \binom{[C]}{L}$.
2) Cyclic wraparound cache access. In this $\mathcal{L}_k = \{k, k+1, \ldots, k+L-1\}$ where addition is modulo C except $k + l = C$ if $k + l$ is multiple of C.

The system operates in two phases.

Placement Phase: In this phase, all C cache nodes are filled. Let Z_c denote the contents stored in cache $c \in [C]$. Z_c is a function of files $W_{[1:N]}$ and all the servers know the contents stored at each of the helper caches.

Private Delivery Phase: In this phase, each user wish to retrieve a file from the servers. User k will choose $d_k \in [N]$ and wish to retrieve W_{d_k} privately from the servers. Let $\mathbf{d} = (d_1, d_2, \ldots, d_K)$ be the demand vector. In order to retrieve their desired files from the servers, users will cooperatively

generate S queries $Q_s^{\mathbf{d}}, s \in [S]$ based on their demands and content stored in helper cache. Then query $Q_s^{\mathbf{d}}$ will be sent to server $s, \forall s \in [S]$. These queries are constructed such that they don't disclose the demand vector $\mathbf{d}$ to any of the server. After receiving their respective queries, server $s, \forall s \in [S]$, will broadcast $A_s^{\mathbf{d}}$ to the users which is a function of the query $Q_s^{\mathbf{d}}$ and $W_{[1:N]}$. After receiving $A_{[S]}^{\mathbf{d}}$, user k, $\forall k \in [K]$, will decode W_{d_k} with the help of the caches it has access to.

Formally we can say that in order to preserve the privacy of demands of the users, the following condition needs to be satisfied:

$$I(\mathbf{d}; Q_s^{\mathbf{d}}, Z_{[1:C]}) = 0, \forall s \in [S].$$

This condition, known as the privacy condition, ensures that none of the servers has any information about user demands. The condition

$$H(W_{d_k}|\mathbf{d}, Z_{\mathcal{L}_k}, A_{[S]}^{\mathbf{d}}) = 0, \forall k \in [K]$$

known as the correctness condition, ensures that users will have no ambiguity about their demanded file.

The rate R is defined to be the amount of data that has to be transmitted by all the servers in order to satisfy the user demand, and is given by

$$R = \sum_{s=1}^{S} H(A_s^{\mathbf{d}}).$$

Our goal is to design placement and private delivery phases jointly that satisfy the privacy and correctness conditions and minimizes the rate.

III. Main Results

In this section, we present the main results of the paper. For a given multi-access cache-aided MuPIR problem, we give a scheme to achieve the rate described in Theorem 1. The scheme is described in Section IV.

Theorem 1. *For Multi-Access Coded caching setup, with S servers, N files, C helper caches and $K = \binom{C}{L}$ users, where each user is accessing a unique set of L helper cache and each cache can store M files and $t = \frac{CM}{N}$ is an integer, the users can retrieve their required file privately i.e. without revealing their demand to any of the servers, with rate*

$$R(t) = \frac{\binom{C}{t+L}}{\binom{C}{t}} \left(1 + \frac{1}{S} + \ldots + \frac{1}{S^{N-1}}\right)$$

Proof. *A scheme, along with the proof of privacy for the scheme, is given in Section IV-B that achieves the rate stated above.* □

Theorem 1 gives an achievable rate in multi-access setups where cache memory M is an integer multiple of N/C. For intermediate memory points, the lower convex envelope of points

$$\{(t, R(t))\}_{t \in [0:C]}$$

can be achieved using memory sharing.

Next, we show that the rate achieved in Theorem 1 is order optimal within a factor of 2 under the assumption of uncoded cache placement.

Theorem 2. *Under the assumption of uncoded cache placement, the rate achieved in Theorem 1 is less than or equal to twice the optimal worst-case rate $R^*(t)$ i.e.*

$$R(t) \leq 2R^*(t).$$

Proof. *The optimal rate achieved in multi-access coded caching without PIR constraint can only be as high as the optimal rate achievable in a multi-access coded caching setup with PIR constraint. So using* (1) *we have,*

$$\begin{aligned} & R^*(t) \geq R^*_{nPIR}(t) \\ \implies & \frac{R^*(t)}{R(t)} \geq \frac{R^*_{nPIR}(t)}{R(t)} \\ \implies & \frac{R(t)}{R^*(t)} \leq \left(1 + \frac{1}{S} + \ldots + \frac{1}{S^{N-1}}\right). \end{aligned}$$

As $\left(1 + \frac{1}{S} + \ldots + \frac{1}{S^{N-1}}\right) \leq 2$ for all $S \geq 2$ we have

$$R(t) \leq 2R^*(t).$$

□

Remark: We have stated that $R(t) \leq 2R^*(t)$, note that the upper bound of $2R^*(t)$ is incurred only when $S = 2$ and $N \to \infty$. For most other cases we have that $R(t) < 2R^*(t)$ e.g. if $S = 10$ and $N = 100$ then $R(t) \approx 1.1R^*_{comb}(t)$ which is only at most 10% higher than optimal rate. And for the special case when $S \to \infty$, optimal rate is achieved.

Before stating the rate for cyclic wraparound cache access setup, we define the quantity $\text{cyc}(n, k, m)$ for integers $m \leq k < n$. $\text{cyc}(n, k, m)$ is the number of k sized subsets of n distinguishable elements arranged in a circle, such that there is atleast one set of m consecutive elements amongst those k elements. Expression for $\text{cyc}(n, k, m)$ is given in (2).

Theorem 3. *For Cyclic Wraparound Multi-Access Coded caching setup, with S servers, N files, K helper caches and K users, where each user is accessing L helper cache in cyclic wraparound manner and each cache can store M files and $t = \frac{KM}{N}$ is an integer, the users can retrieve their required file privately i.e. without reveling their demand to any of the servers, with rate*

$$R(t) = \min\{\frac{K-t}{t+1} R^*_{PIR}(S, N), R'(t)\}, \text{ where} \quad (3)$$

$$R'(t) = \frac{cyc(K, t+L, L)}{\binom{K}{t}} R^*_{PIR}(S, N). \quad (4)$$

Proof. *In Section IV-D we have given an achievable scheme that achieve rate $R'(t)$ as stated above for cyclic wraparound cache access setup.* □

Theorem 3 gives an achievable rate in Multi-Access setup where cache memory M is an integer multiple of N/K. For intermediate memory points, lower convex envelope of points

$$\{(t, R(t))\}_{t \in [0:K]}$$

can be achieved by memory sharing.

$$
\begin{aligned}
cyc(n,k,m) = &\sum_{r=1}^{k}\left(\binom{n-k}{r}+\binom{n-k-1}{r-1}\right)\sum_{l=1}^{r}(-1)^{l-1}\binom{r}{l}\binom{k-l(m-1)-1}{r-1}\\
&+\sum_{r=3}^{k}\binom{n-k-1}{r-2}\Big(\sum_{l=2}^{m-1}(l-1)\sum_{j=1}^{r-2}(-1)^{j-1}\binom{r-2}{j}\binom{k-l-j(m-1)-1}{r-3}\\
&+\sum_{l=m}^{k}(l-1)\binom{k-l-1}{r-3}\Big)+k-1
\end{aligned}
\tag{2}
$$

A. *Comparison with dedicated cache setup of [7]*

We will compare our scheme with the Product Design given in [7]. In dedicated cache setup of [7] there are N files $\{W_n\}_{n\in[N]}$ replicated across S servers. There are K users, each equipped with a dedicated cache capable of storing M files. Users want to privately retrieve their desired files from the servers. In [7] an achievable scheme called *Product Design* is given that achieves rate R_{PD} given by

$$R_{PD} = \frac{K-t}{t+1}\left(1+\frac{1}{S}+\ldots+\frac{1}{S^{N-1}}\right)$$

where $t = KM/N$. Note that the rate achieved by the product design is the same as the rate achieved in Theorem 1 for the special case of $L = 1$ i.e. when every user is accessing only one file.

In coded caching systems, the number of cache nodes and the storage capacity of each cache node are crucial parameters. In dedicated cache systems, the number of cache nodes and the number of users supported in the networks are the same. In contrast, multi-access coded caching systems can support more number of users for the same number of cache nodes. So, in multi-access systems, even if the number of transmissions is more than that of a dedicated cache system, it is possible that one transmission is beneficial to more number of users. So we will be considering the parameter *per user rate* i.e. $\frac{R}{K}$ for comparing two systems. For distinction, quantities related to a dedicated cache system will have subscript DC, and multi-access setup quantities will have subscript MA. We compare our scheme with the product design in the following three settings with 2 servers and 3 files.

- Both dedicated cache system and multi-access system have the same number of caches, i.e. C cache nodes in both settings, and cache size is also the same in both settings, i.e. $M_{DC} = M_{MA} = M$. In this case, there will be C users in the dedicated cache setup and $\binom{C}{L}$ users in the multi-access setup.
- Both dedicated cache system and multi-access system have the same number of caches, i.e. C cache nodes in both settings, but each user is accessing the same amount of memory. As users in the multi-access system are accessing L cache nodes each of size M_{MA} and in the dedicated cache system, each user is accessing only one cache of size M_{DC} we will set $M_{DC} = L \times M_{MA}$. In this case, also, there will be C users in the dedicated cache setup and $\binom{C}{L}$ users in the multi-access setup.

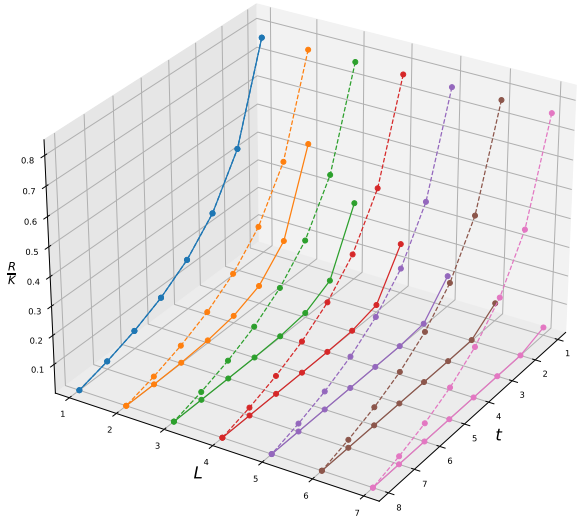

Fig. 2: $\frac{R}{K}$ for Dedicated cache (dotted lines) and Multi Access (solid lines). Here $C_{MA} = C_{DC} = 8$ and $t_{MA} = t_{DC} = t$

- Number of users in both systems are the same, i.e. $K_{MA} = K_{DC}$ and total system memory is also the same. Considering C cache nodes in multi-access system we have $K_{MA} = \binom{C}{L} = K_{DC}$ and as the number of cache nodes in the dedicated cache system is the same as the number of users, there will be $\binom{C}{L}$ cache nodes in the dedicated cache system. For same total memory in both settings, we want $M_{DC} \times \binom{C}{L} = M_{MA} \times C$.

Also, note that the parameter $t = \frac{CM}{N}$ in multi-access setup and $t = \frac{KM}{N}$ in dedicated cache setup denote how many times the entire set of N files can be replicated across the cache. For instance, if $t = 2$ then cache nodes are capable of storing $2N$ units. Also, total memory of the system is tN units, which is equal to KM for dedicated cache systems and CM for multi-access setups. For dedicated cache systems, the number of cache nodes is always equal to the number of users K.

1) Same number of cache and same amount of memory: As the number of cache nodes in a multi-access system is C_{MA} and in a dedicated cache system, the number of cache nodes are same as the number of users K_{DC}, we consider $C_{MA} = K_{DC} = 8$ i.e. 8 cache nodes in both, dedicated cache and multi-access setup. We also consider total memory of both systems are also same i.e. $t_{MA} = t_{DC} = t$. Let the

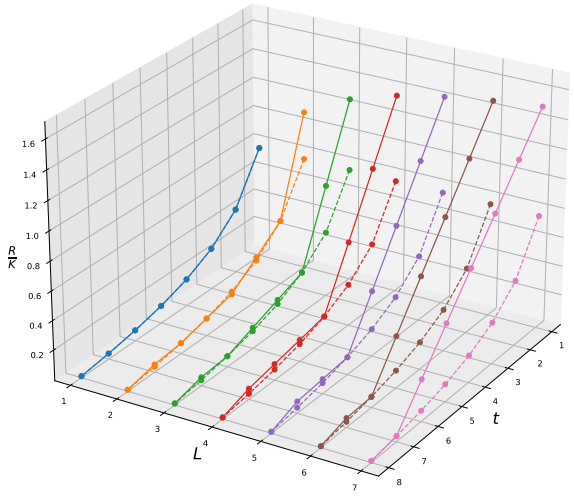

Fig. 3: $\frac{R}{K}$ for Dedicated cache (dotted lines) and Multi Access (solid lines). Here $C_{MA} = C_{DC} = 8$ and $t_{DC} = Lt_{MA} = t$

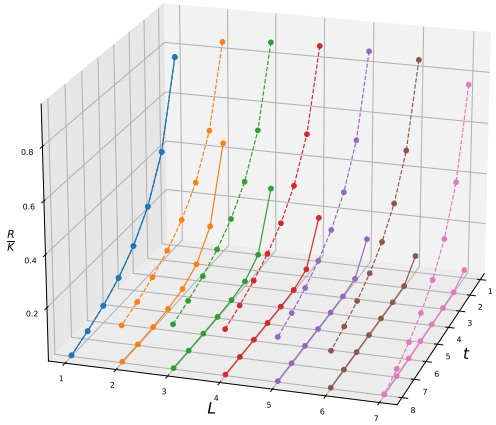

Fig. 4: $\frac{R}{K}$ for Dedicated cache (dotted lines) and Multi Access (solid lines). Here $C_{MA} = 8$, $K_{MA} = K_{DC} = \binom{C_{MA}}{L}$ and $t_{DC} = t_{MA} = t$

cache access degree for the multi-access system be L. Note that $K_{MA} = \binom{C}{L}$, i.e. the number of users in the multi-access setup will be higher than the dedicated cache system except for $L = 1$ where both systems are the same. So with the same amount of memory, multi-access setup is able to support a higher number of users. We will compare the rate per user for the two setups. Note that although $t_{MA} = t_{DC}$, users in Multi-Access setup are accessing L cache nodes. Therefore they have access to more cache memory than users of the dedicated cache setting. Rate per user for both systems is plotted in Figure 2 for 8 cache nodes in each setting for different values of t and L. We can see that per-user rate of multi-access setup is better than that of dedicated cache setup. Due to large number of users in multi-access setup, a single transmission from a server can benefit more users than the number of users benefited by a single transmission in a dedicated cache setup. We will see that in a multi-access system, a single transmission is simultaneously used by $\binom{L+t}{L}$ users, compared to $t+1$ in a dedicated cache setup.

2) Same number of cache and same memory per user: As we saw, users in multi-access setup are accessing more memory than users of dedicated cache setting if $C_{MA} = K_{DC}$ and $t_{MA} = t_{DC}$. Now, we will reduce the storage capacity in a multi-access setup so that the amount of memory accessed by users of dedicated cache setup and multi-access setup is the same. We consider that $C_{MA} = K_{DC} = 8$, but this time storage capacity of cache nodes in a multi-access setup is smaller than that of a dedicated cache system so that each user is accessing the same amount of memory. For that we set $M_{DC} = L \times M_{MA}$ because every user in multi-access setup is accessing L cache nodes. Therefore we have

$$\frac{t_{DC}N}{K_{DC}} = L\frac{t_{MA}N}{C_{MA}} \implies t_{MA} = t_{DC}/L.$$

Now, although each user has access to the same amount of memory, each user of the dedicated cache system has access to a cache whose content is independent of content stored in caches of other users. Whereas cache content accessed by the users of multi-access setup is not independent of cache content accessed by other users, therefore the rate of dedicated cache setup will be better than the rate of multi-access setup. In Figure 3 we compare per user rate for L and t_{DC} ranging in $[C]$. Although even for the per-user rate we see that the dedicated system is performing better than the multi-access setup for most of the cases, we see that for cases when $t_{DC} = L$ per user rate of both settings coincide as

$$\frac{R_{MA}/K_{MA}}{R_{DC}/K_{DC}} = \frac{\frac{1}{\binom{C}{L}}\frac{\binom{C}{L+1}}{C}}{\frac{1}{C}\frac{\binom{C}{t_{DC}+1}}{\binom{C}{t_{DC}}}} = 1.$$

Beyond that, there also exist some points where, per user rate of the multi-access setup is better than that of dedicated cache setup, for instance in Figure 3 when $t_{DC} = 4$ and $L = 2$ we have $\frac{R_{DC}}{K_{DC}} = \frac{1}{10}$ whereas $\frac{R_{MA}}{K_{MA}} = \frac{1}{11.2}$.

3) Same number of users and same total system memory: We keep the number of users in both setup same. So $K_{DC} = K_{MA} = \binom{C_{MA}}{L}$. Also we keep total memory in both systems same i.e. $t_{MA} = t_{DC} = t$. Note that the number of cache nodes in dedicated cache system is $K_{DC} = \binom{C_{MA}}{L}$. For same total memory, we want that

$$C_{MA}M_{MA} = K_{DC}M_{DC} \implies M_{DC} = \frac{C_{MA}}{\binom{C_{MA}}{L}}M_{MA}.$$

So, the size of individual cache nodes in the dedicated cache system is now smaller than that of the multi-access system. Then again, users of multi-access setup can access more than one cache node. Therefore, we see that dedicated cache system has two disadvantages. For supporting same number of users as multi-access setup, using same total system memory, the size of the individual cache have to be reduced, and every

user gets a smaller share of total memory compared to a multi-access system. The comparison for the per-user rate for this setting is shown in Figure 4. We can see that the multi-access system is performing better than the dedicated cache setup while utilising the same amount of cache memory and serving the same number of users.

IV. Scheme Description

A. Example

We first describe the proposed scheme with an example. Consider multi-access setup with $S = 2$ non-colluding servers, $N = 3$ files W_1, W_2 and W_3. There are $C = 5$ cache nodes, each node capable of storing $M/N = 2/5$ fraction of each file. There are $K = 10$ users and each user is connected to a unique set of $L = 3$ cache nodes. We index the users with the indices of cache nodes they are connected to. For example, user $\{2, 3, 4\}$ is connected to cache node 2, cache node 3 and cache node 4. The caches are filled in the placement phase as follows:

Placement Phase: Let $t = \frac{CM}{N} = 2$. Divide each file into $\binom{C}{t} = \binom{5}{2} = 10$ subfiles.

$$W_n = \left\{ W_{n,\mathcal{T}} | \mathcal{T} \in \binom{[5]}{2} \right\}$$

and fill cache node $c \in [5]$ as follows:

$$Z_c = \left\{ W_{n,\mathcal{T}} | n \in [N], \forall \mathcal{T} \in \binom{[5]}{2} \text{ such that } c \in \mathcal{T} \right\}.$$

Delivery Phase: Now every user choose a file index, and want to retrieve the file from the servers privately. Let the demand of user $\mathcal{K} \in \binom{[5]}{2}$ be $d_{\mathcal{K}}$. Then the demand vector is $\mathbf{d} = (d_{\mathcal{K}})_{\mathcal{K} \in \binom{[5]}{3}}$, and users want to hide this demand vector from the servers. For this, each subfile is further divided into $S^N = 8$ sub-subfiles and the users will generate 2 queries $\mathbf{Q}_1^{\mathbf{d}}$ and $\mathbf{Q}_2^{\mathbf{d}}$ one for each server as follows. For every $\mathcal{S} \in \binom{[C]}{t+L} = \binom{[5]}{5} = \{\{1, 2, 3, 4, 5\}\}$ generate

$$\mathbf{Q}_s^{\mathbf{d},\mathcal{S}} = \left\{ Q_s^{d_{\mathcal{K}},\mathcal{S}} | \mathcal{K} \in \binom{\mathcal{S}}{3} \right\}$$

where $Q_s^{d_{\mathcal{K}},\mathcal{S}}$ is the query sent to server s in single user PIR setup, with S servers and N files if the demand of the user is $d_{\mathcal{K}}$. For instance, consider $\mathcal{K} = \{1, 2, 3\}$ and $d_{\{1,2,3\}} = 1$ then in order to generate $Q_1^{1,\mathcal{S}}, Q_2^{1,\mathcal{S}}$ three random permutations of $[S^N] = 8$ will be formed, one corresponding to each file. Let these permutations be $\{a_1 \dots a_8\}, \{b_1 \dots b_8\}$ and $\{c_1 \dots c_8\}$ for files W_1, W_2 and W_3 respectively. Now, as the user $\{1, 2, 3\}$ wants file W_1 then the query $Q_s^{1,\mathcal{S}}$ will be a list of sub-subfile index of subfiles $\{4, 5\}$ as given in Table I. A list will be generated for every $\mathcal{K} \in \binom{[5]}{3}$ and queries will be sent to respective servers. After receiving the query server s will transmit

$$\bigoplus_{\mathcal{K} \in \binom{\mathcal{S}}{3}} A_s^{d_{\mathcal{K}}}(Q_s^{d_{\mathcal{K}},\mathcal{S}}, W_{[3],\mathcal{S}\setminus\mathcal{K}})$$

where $A_s^{d_{\mathcal{K}}}(Q_s^{d_{\mathcal{K}},\mathcal{S}}, W_{[3],\mathcal{S}\setminus\mathcal{K}})$ is the answer of server s in single user PIR setup if query is $Q_s^{d_{\mathcal{K}},\mathcal{S}}$ and the set of files is $W_{[3],\mathcal{S}\setminus\mathcal{K}}$. Again, considering $\mathcal{K} = \{1, 2, 3\}$, $A_s^1(Q_s^{1,\mathcal{S}}, W_{[3],\mathcal{S}\setminus\{1,2,3\}})$ is given in Table II. After listening to the broadcast from the servers, every user will be able to decode their desired subfile. Again considering the case of the user $\{1, 2, 3\}$, it has access to all subfiles of all files indexed by $\{1, 2\}, \{1, 3\}, \{1, 4\}, \{1, 5\}, \{2, 3\}, \{2, 4\}, \{2, 5\}$ and $\{3, 4\}$. Only missing subfiles are those indexed by $\{4, 5\}$. Now consider again the transmission of the server

$$\begin{aligned}&\bigoplus_{\mathcal{K} \in \binom{\mathcal{S}}{3}} A_s^{d_{\mathcal{K}}}(Q_s^{d_{\mathcal{K}}}, W_{[3],\mathcal{S}\setminus\mathcal{K}}) \\ =&A_s^1(Q_s^1, W_{[3],\{4,5\}}) + \bigoplus_{\mathcal{K} \in \binom{\mathcal{S}}{3}\setminus\{1,2,3\}} A_s^{d_{\mathcal{K}}}(Q_s^{d_{\mathcal{K}}}, W_{[3],\mathcal{S}\setminus\mathcal{K}}).\end{aligned}$$

User $\{1, 2, 3\}$ has access to all the subfiles indexed by $\mathcal{S} \setminus \mathcal{K}, \forall \mathcal{K} \in \binom{\mathcal{S}}{3} \setminus \{1, 2, 3\}$, so it can cancel out the summation term above and be left with $A_1^1(Q_s^{1,\mathcal{S}}, W_{[3],\{4,5\}})$ and $A_2^1(Q_s^{1,\mathcal{S}}, W_{[3],\{4,5\}})$. The user $\{1, 2, 3\}$ will get all sub-subfiles of $W_{1,\{4,5\}}$ from these remaining terms. Similarly all users will get the missing subfiles of the demanded file. Also, to generate $\{Q_s^{d_{\mathcal{K}},\mathcal{S}}\}_{s\in[S]}$ independent random permutations of $[8]$ is chosen for every $\mathcal{S}$. Then, from the privacy of singel user PIR scheme, servers will get no information about the demand vector from the queries they got.

Also note that, as each server is transmitting 7 sub-subfiles each of size $\frac{1}{10\times 8}$ units, the rate in this example is $R = \frac{7}{40}$ and subpacketization level is 80.

B. General Description

Consider N files $\{W_n\}_{n\in[N]}$ replicated across S servers. There are C cache nodes, each capable of storing M files and K users, each connected to a unique set of L cache nodes. As each user is connected to a unique set of L cache nodes, we index each user with an L sized subset of $[C]$. Specifically, user $\mathcal{K}$, where $\mathcal{K} \in \binom{[C]}{L}$, is the user connected to the cache nodes indexed by $\mathcal{K}$. Let $\mathcal{U}$ be the set of all users where

$$\mathcal{U} \in \binom{\binom{[C]}{L}}{K}.$$

Placement Phase: Let $t = \frac{CM}{N}$ be an integer. Then divide each file into $\binom{C}{t}$ subfiles, each indexed by a t sized subset of $[C]$.

$$W_n = \left\{ W_{n,\mathcal{T}} | \mathcal{T} \in \binom{[C]}{t} \right\}.$$

Then fill cache node c with

$$Z_c = \left\{ W_{n,\mathcal{T}} | c \in \mathcal{T}, \mathcal{T} \in \binom{[C]}{t} \right\}.$$

Delivery Phase: In this phase, every user chooses one of the file indices. Let user $\mathcal{K}$ chooses $d_{\mathcal{K}} \in [N], \forall \mathcal{K} \in \mathcal{U}$, then $\mathbf{d} = (d_{\mathcal{K}})_{\mathcal{K}\in\mathcal{U}}$ is the demand vector. Users don't want the servers to get any information about the demand vector. For privately retrieving the files, users cooperatively generate S queries $\mathbf{Q}_s^{\mathbf{d}}, \forall s \in [S]$. For every $\mathcal{S} \in \binom{[C]}{t+L}$, such that $\mathcal{S} \supset \mathcal{K}$ for at least one $\mathcal{K} \in \mathcal{U}$, users generate

$$\mathbf{Q}_s^{\mathbf{d},\mathcal{S}} = \left\{ Q_s^{d_{\mathcal{K}},\mathcal{S}} \middle| \mathcal{K} \in \binom{\mathcal{S}}{L} \cap \mathcal{U} \right\}.$$

TABLE I: Query Table

Server 1	Server 2
a_1, b_1, c_1	a_2, b_2, c_2
a_3, b_2	a_4, b_1
b_3, c_3	b_4, c_4
a_5, c_2	a_6, c_1
a_7, b_4, c_4	a_8, b_3, c_3

TABLE II: Answer from servers

Server 1	Server 2
$W^{a_1}_{1,\{4,5\}}, W^{b_1}_{2,\{4,5\}}, W^{c_1}_{3,\{4,5\}}$	$W^{a_2}_{1,\{4,5\}}, W^{b_2}_{2,\{4,5\}}, W^{c_2}_{3,\{4,5\}}$
$W^{a_3}_{1,\{4,5\}} + W^{b_2}_{2,\{4,5\}}$ $W^{b_3}_{2,\{4,5\}} + W^{c_3}_{3,\{4,5\}}$ $W^{a_5}_{1,\{4,5\}} + W^{c_2}_{3,\{4,5\}}$	$W^{a_4}_{1,\{4,5\}} + W^{b_1}_{2,\{4,5\}}$ $W^{b_4}_{2,\{4,5\}} + W^{c_4}_{3,\{4,5\}}$ $W^{a_6}_{1,\{4,5\}} + W^{c_1}_{3,\{4,5\}}$
$W^{a_7}_{1,\{4,5\}} + W^{b_4}_{2,\{4,5\}} + W^{c_4}_{3,\{4,5\}}$	$W^{a_8}_{1,\{4,5\}} + W^{b_3}_{2,\{4,5\}} + W^{c_3}_{3,\{4,5\}}$

Here $Q_s^{d_\mathcal{K},\mathcal{S}}$ is the query sent to server s in the single user PIR setup of [2] if the user demand is $d_\mathcal{K}$. The query sent to the server s is

$$\mathbf{Q}_s^{\mathbf{d}} = \left\{ \mathbf{Q}_s^{\mathbf{d},\mathcal{S}} \middle| \mathcal{S} \in \binom{[C]}{t+L}, \mathcal{S} \supset \mathcal{K} \text{ for some } \mathcal{K} \in \mathcal{U} \right\}.$$

Now for every $\mathbf{Q}_s^{\mathbf{d},\mathcal{S}}$, server s will transmit

$$\bigoplus_{\mathcal{K} \in \binom{\mathcal{S}}{L} \cap \mathcal{U}} A_s^{d_\mathcal{K}}(Q_s^{d_\mathcal{K},\mathcal{S}}, W_{[N],\mathcal{S}\setminus\mathcal{K}})$$

where $A_s^{d_\mathcal{K}}(Q_s^{d_\mathcal{K},\mathcal{S}}, W_{[N],\mathcal{S}\setminus\mathcal{K}})$ is the answer of server s in the single user PIR setup if the received query is $Q_s^{d_\mathcal{K},\mathcal{S}}$ and the set of files is $\{W_{[N],\mathcal{S}\setminus\mathcal{K}}\}$.

Decoding: Consider user $\mathcal{K}$ (i.e. the user connected to cache nodes indexed by $\mathcal{K}$) and a subfile index $\mathcal{T}$. If $\mathcal{K} \cap \mathcal{T} \neq \phi$ then subfile $W_{d_\mathcal{K},\mathcal{T}}$ is available to the user from the cache nodes. If $\mathcal{K} \cap \mathcal{T} = \phi$ then the subfile has to be decoded from the transmissions. Consider transmissions corresponding to $\mathcal{S} = \mathcal{K} \cup \mathcal{T}$

$$\begin{aligned}
&\bigoplus_{\mathcal{K}' \in \binom{\mathcal{K}\cup\mathcal{T}}{L} \cap \mathcal{U}} A_s^{d_{\mathcal{K}'}}(Q_s^{d_\mathcal{K},\mathcal{K}\cup\mathcal{T}}, W_{[N],(\mathcal{K}\cup\mathcal{T})\setminus\mathcal{K}'}) \\
&= A_s^{d_\mathcal{K}}(Q_s^{d_\mathcal{K},\mathcal{K}\cup\mathcal{T}}, W_{[N],\mathcal{T}}) \oplus \\
&\quad \bigoplus_{\mathcal{K}' \in \binom{\mathcal{K}\cup\mathcal{T}}{L} \cap \mathcal{U}\setminus\mathcal{K}} A_s^{d_{\mathcal{K}'}}(Q_s^{d_{\mathcal{K}'},\mathcal{K}\cup\mathcal{T}}, W_{[N],(\mathcal{K}\cup\mathcal{T})\setminus\mathcal{K}'}).
\end{aligned}$$

User $\mathcal{K}$ has access to all the subfiles in the second term of the RHS above, and therefore it can recover the first term from the above expression. After getting $A_s^{d_\mathcal{K}}(Q_s^{d_\mathcal{K},\mathcal{K}\cup\mathcal{T}}, W_{[N],\mathcal{T}})$ for all $s \in [S]$, user $\mathcal{K}$ can recover $W_{d_\mathcal{K},\mathcal{T}}$ using the single user PIR retrieval scheme [2].

Proof of Privacy: Consider a $Q_s^{d_\mathcal{K},\mathcal{S}} \in \mathbf{Q}_s^{\mathbf{d},\mathcal{S}}$ for some $\mathbf{Q}_s^{\mathbf{d},\mathcal{S}} \in \mathbf{Q}_s^{\mathbf{d}}$. From privacy of single user PIR scheme we know that $Q_s^{d_\mathcal{K},\mathcal{S}}$ is independent of the demand of user $\mathcal{K}$ i.e. $d_\mathcal{K}$. Also $Q_s^{d_\mathcal{K},\mathcal{S}}$ is independent of the demands of other users because an independent and random permutation of $[S^N]$ is used to construct $Q_s^{d_\mathcal{K},\mathcal{S}}$. This is true for every $Q_s^{d_\mathcal{K},\mathcal{S}} \in \mathbf{Q}_s^{\mathbf{d},\mathcal{S}}$ for every $\mathbf{Q}_s^{\mathbf{d},\mathcal{S}} \in \mathbf{Q}_s^{\mathbf{d}}$. Also $Q_s^{d_\mathcal{K},\mathcal{S}}$ for all $\mathcal{K}$ and for all $\mathcal{S}$ are mutually independent because random and independent permutations of $[S^N]$ are used for every $Q_s^{d_\mathcal{K},\mathcal{S}}$ and every user is choosing a desired file index independently. This completes the proof.

C. Multi Access Setup when $K = \binom{C}{L}$

Here we characterize the performance of the scheme given in Section IV-B for the case when $K = \binom{C}{L}$.

Rate: Each server performs $\binom{C}{t+L}$ transmissions corresponding to every $\mathcal{S} \in \binom{[C]}{L}$ and each transmission is of size $\frac{1}{\binom{C}{t}}\left(\frac{1}{S} + \frac{1}{S^2} + \ldots + \frac{1}{S^N}\right)$ units. So the rate of the scheme is

$$R(t) = \frac{\binom{C}{t+L}}{\binom{C}{t}}\left(1 + \frac{1}{S} + \frac{1}{S^2} + \ldots + \frac{1}{S^{N-1}}\right).$$

Subpacketization: Each file is divided into $\binom{C}{t}$ subfiles, each of which is further divided into S^N sub-subfiles. So the subpacketization level is $\binom{C}{t} \times S^N$.

Coding Gain: The transmission corresponding to each $\mathcal{S} \in \binom{[C]}{t+L}$ is beneficial to user $\mathcal{K}$ if $\mathcal{K} \in \binom{\mathcal{S}}{L}$. So every transmission, from each server, is used by $\binom{L+t}{L}$ users.

D. Cyclic wraparound cache access

Now consider a cyclic wraparound access setup with C users and C cache nodes. User k is accessing cache nodes indexed by $\mathcal{K}_k = \{k, k+1, \ldots, k+L-1\}$ where addition is modulo C except $k+l = C$ if $k+l$ is a multiple of C. In this case, transmissions are performed only for those $\mathcal{S} \in \binom{[C]}{t+L}$ which contain L consecutive integers (with wrapping around C allowed). This is same as the number of ways of choosing $t+L$ elements from a set of C distinguishable elements arranged in a circle, which contain atleast one subset of L consecutive elements. It is shown in [17], that there are $cyc(C, t+L, L)$ ways, as described in (2), to choose $t+L$ caches out of C, in such a manner. Therefore in cyclic wraparound cache access setup, only $cyc(C, t+L, L)$ transmissions are required.

Also note that, user k of the dedicated cache setup as well as user k of the cyclic wraparound cache access setup are accessing cache node k. In dedicated cache setup $\binom{C}{t+1}$ transmissions, each of size $\frac{1}{\binom{C}{t}}$ units, are performed [7]. Therefore, when $cyc(C, t+L, L) > \binom{C}{t+1}$, transmissions of dedicated cache setup are performed. In this scenario the rate achieved in multi-access setup will only be as high as the rate achieved in the dedicated cache scenario with same cache sizes. For $t \in [0 : C]$ the rate achieved by the cyclic wraparound setup is

$$\min\left\{\frac{C-t}{t+1}, \frac{cyc(C, t+L, L)}{\binom{C}{t}}\right\}\left(1 + \frac{1}{S} + \ldots + \frac{1}{S^{N-1}}\right).$$

We demonstrate our point using an example with $C = 8$ and $L = 2$. In Figure 5 we see that for smaller values of t the cyclic wraparound cache access system is incurring more per user rate than the dedicated cache setup. For instance, when $t = 2$, $cyc(8, 4, 2) = 68$ transmissions are performed for

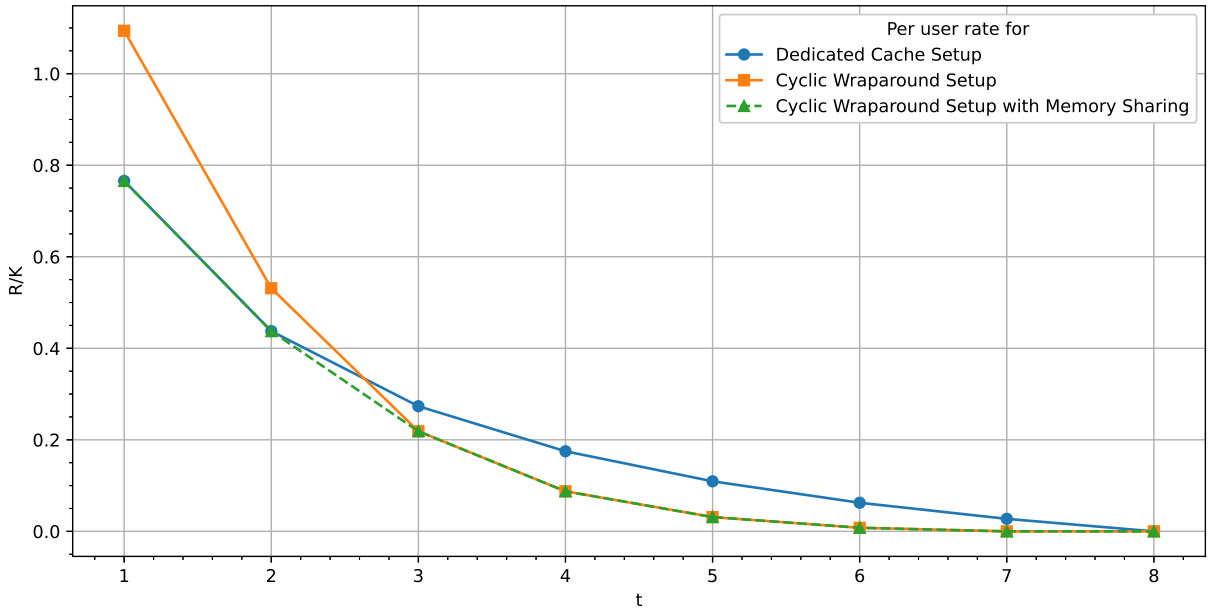

Fig. 5: Per user rate for $C = 8, L = 2, S = 2, N = 3$. Multi Access setup with cyclic wraparound cache access achieve rate only as high as dedicated cache setup with equal total memory in both systems.

the cyclic wraparound cache access without memory sharing (and incurring per user rate 0.531) compared to $\binom{8}{3} = 56$ transmissions in dedicated cache setup (and incurring per user rate 0.437). Therefore when $t = 2$, transmissions corresponding to the dedicated cache setup are performed. But when $t = 3$, the cyclic wraparound cache access setup satisfy user demands with 56 transmissions (and incurring per user rate 0.219) compared to the dedicated cache setup which require 70 transmissions (and incur per user rate 0.273), and therefore transmissions as described in Section IV-B are performed.

Acknowledgement

This work was supported partly by the Science and Engineering Research Board (SERB) of Department of Science and Technology (DST), Government of India, through J.C. Bose National Fellowship to B. Sundar Rajan, and by the Ministry of Human Resource Development (MHRD), Government of India, through Prime Minister's Research Fellowship (PMRF) to Kanishak Vaidya.

References

[1] B. Chor, O. Goldreich, E. Kushilevitz and M. Sudan, "Private information retrieval", Proceedings of IEEE 36th Annual Foundations of Computer Science, 1995, pp. 41-50, doi: 10.1109/SFCS.1995.492461.

[2] H. Sun and S. A. Jafar, "The Capacity of Private Information Retrieval", in IEEE Transactions on Information Theory, vol. 63, no. 7, pp. 4075-4088, July 2017, doi: 10.1109/TIT.2017.2689028.

[3] H. Sun and S. A. Jafar, "The Capacity of Robust Private Information Retrieval With Colluding Databases", in IEEE Transactions on Information Theory, vol. 64, no. 4, pp. 2361-2370, April 2018, doi: 10.1109/TIT.2017.2777490.

[4] H. Y. Lin, S. Kumar, E. Rosnes, A. G. i. Amat and E. Yaakobi, "Weakly-Private Information Retrieval", 2019 IEEE International Symposium on Information Theory (ISIT), 2019, pp. 1257-1261, doi: 10.1109/ISIT.2019.8849444.

[5] Z. Chen, Z. Wang and S. A. Jafar, "The Capacity of T-Private Information Retrieval With Private Side Information", in IEEE Transactions on Information Theory, vol. 66, no. 8, pp. 4761-4773, Aug. 2020, doi: 10.1109/TIT.2020.2977919.

[6] 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, doi: 10.1109/TIT.2014.2306938.

[7] X. Zhang, K. Wan, H. Sun, M. Ji and G. Caire, "On the Fundamental Limits of Cache-Aided Multiuser Private Information Retrieval", in IEEE Transactions on Communications, vol. 69, no. 9, pp. 5828-5842, Sept. 2021, doi: 10.1109/TCOMM.2021.3091612.

[8] P. N. Muralidhar, D. Katyal and B. S. Rajan, "Maddah-Ali-Niesen Scheme for Multi-access Coded Caching", 2021 IEEE Information Theory Workshop (ITW), 2021, pp. 1-6, doi: 10.1109/ITW48936.2021.9611394.

[9] F. Brunero and P. Elia, "Fundamental limits of combinatorial multi-access caching", arXiv:2110.07426, 14 Oct 2021.

[10] D. Katyal, P. N. Muralidhar and B. S. Rajan, "Multi-Access Coded Caching Schemes From Cross Resolvable Designs", in IEEE Transactions on Communications, vol. 69, no. 5, pp. 2997-3010, May 2021, doi: 10.1109/TCOMM.2021.3053048.

[11] J. Hachem, N. Karamchandani and S. N. Diggavi, "Coded Caching for Multi-level Popularity and Access", in IEEE Transactions on Information Theory, Vol.63, no.5, pp.3108-3141.

[12] K. S. Reddy and N. Karamchandani, "Rate-Memory Trade-off for Multi-Access Coded Caching With Uncoded Placement", in IEEE Transactions on Communications, vol. 68, no. 6, pp. 3261-3274, June 2020, doi: 10.1109/TCOMM.2020.2980817.

[13] P. Trinadh, M. Dutta, A. Thomas and B. S. Rajan, "Decentralized Multi-access Coded Caching with Uncoded Prefetching", 2021 IEEE Information Theory Workshop (ITW), 2021, pp. 1-6, doi: 10.1109/ITW48936.2021.9611497.

[14] M. Cheng, K. Wan, D. Liang, M. Zhang and G. Caire, "A Novel Transformation Approach of Shared-Link Coded Caching Schemes for Multiaccess Networks", in IEEE Transactions on Communications, vol. 69, no. 11, pp. 7376-7389, Nov. 2021, doi: 10.1109/TCOMM.2021.3104035.

[15] S. Shanuja and B. Sundar Rajan, "An improved multi-access coded caching with uncoded placement", arXiv:2009.05377v3, 12 Feb 2021.

[16] B. Serbetci, E. Parrinello and P. Elia, "Multi-access coded caching: gains beyond cache-redundancy", 2019 IEEE Information Theory Workshop (ITW), 2019, pp. 1-5, doi: 10.1109/ITW44776.2019.8989128.

[17] K. Vaidya and B. S. Rajan, "Multi-User Private Information Retrieval using Multi Access Coded Caching", arXiv:2201.11481v2, 9 Feb 2022.