entropy

MDPI

Article

Multi-User PIR with Cyclic Wraparound Multi-Access Caches †

Kanishak Vaidya and Balaji Sundar Rajan *

Department of Electrical Communication Engineering, IISc Bangalore, Bengaluru 560012, India; kanishakv@iisc.ac.in

* Correspondence: bsrajan@iisc.ac.in

† Part of the content of this manuscript appears in the conference proceeding of 6th Caching, Computing and Delivery in Wireless Networks Workshop (CCDWN 2022) colocated with the 20th International Symposium on Modeling and Optimization in Mobile, Ad hoc, and Wireless Networks (WiOpt 2022), Turin, Italy, 19 September 2022.

Abstract: We consider the problem of multi-access cache-aided multi-user Private Information Retrieval (MACAMuPIR) with cyclic wraparound cache access. In MACAMuPIR, several files are replicated across multiple servers. There are multiple users and multiple cache nodes. When the network is not congested, servers fill these cache nodes with the content of the files. During peak network traffic, each user accesses several cache nodes. Every user wants to retrieve one file from the servers but does not want the servers to know their demands. This paper proposes a private retrieval scheme for MACAMuPIR and characterizes the transmission cost for multi-access systems with cyclic wraparound cache access. We formalize privacy and correctness constraints and analyze transmission costs. The scheme outperforms the previously known dedicated cache setup, offering efficient and private retrieval. Results demonstrate the effectiveness of the multi-access approach. Our research contributes an efficient, privacy-preserving solution for multi-user PIR, advancing secure data retrieval from distributed servers.

Keywords: coded caching; private information retrieval; multi-access caches

Citation: Vaidya, K.; Rajan, B.S. Multi-User PIR with Cyclic Wraparound Multi-Access Caches. *Entropy* **2023**, *25*, 1228. https://doi.org/10.3390/e25081228

Academic Editor: Songze Li

Received: 2 May 2023
Revised: 4 August 2023
Accepted: 8 August 2023
Published: 18 August 2023

1. Introduction

The problem of Private Information Retrieval (PIR), initially introduced in Chor et al. (1995) [1], revolves around the confidential retrieval of data from distributed servers. Users aim to retrieve a specific file from a collection of files stored across these servers while keeping the servers unaware of the file's identity. Sun et al. (2017) [2] present a PIR scheme that minimizes the user's download cost. Subsequently, the PIR problem has been addressed in various other settings. For instance, in [3], PIR is studied with colluding servers, and in [4], weakly private information retrieval is studied where some information about the user demand is allowed to be known to the servers. In [5], the user is allowed to have files stored as side information.

Currently, PIR is being explored in conjunction with coded caching for content delivery scenarios. As first proposed in [6], the coded caching problem includes a number of users, each with their own cache memories and a single server hosting a number of files. Users fill their caches while the network is not busy, and during periods of high network traffic, they request files from the server. The server broadcasts coded transmissions that benefit multiple users simultaneously. Users can use the content in their caches to decode the files they requested after receiving the broadcasts. A cache-aided PIR technique was recently put up by Caire et al. [7], in which many users, each with access to their own dedicated caches, attempt to privately recover files from non-colluding servers. The advantages of coded caching from [2,6] are combined to provide an order-optimal MuPIR strategy.

In this paper, we use a variation of coded caching known as multi-access coded caching in PIR. In multi-access coded caching, users do not have access to dedicated caches. Instead,

there are helper cache nodes, which are accessed by the users. Multiple users can access one helper cache, and a user can access multiple caches. This paper uses a multi-access setup with cyclic wraparound cache access. In cyclic wraparound cache access, the number of users and cache nodes are equal. Multi-access systems with cyclic wraparound cache access are widely studied in the literature. In [8], Hachem et al. derive an order-optimal caching scheme which judiciously shares cache memory among files with different popularities. This idea was extended to a multi-access setup with cyclic-wraparound cache access. In [9], Reddy et al. studied a multi-access coded caching design and proposed a new achievable rate within a multiplicative gap of at most 2 compared to the lower bound for the said problem provided uncoded placement. In [10], a delivery scheme is proposed for the decentralized multi-access coded caching problem where each user is connected to multiple consecutive caches in a cyclic wrap-around fashion. A lower bound on the delivery rate is also obtained for the decentralized multi-access coded caching problem using techniques from index coding. In [11], Cheng et al. propose a transformation approach to generalize the MAN scheme to the multi-access caching systems, such that the results of [8] remain achievable in full generality. In [12], the authors generalize one of the cases in [13], which proposes novel caching and coded delivery schemes maximizing the local caching gain.

Notation 1. *Consider integers a and b. Then,* $[a:b] \triangleq \{n \in \mathbb{Z} | a \leq n \leq b\}$. $[a] \triangleq [1:a]$. *For a set* $\mathcal{S}$ *of size* $|\mathcal{S}|$ *and an integer* $N \leq |\mathcal{S}|$, $\binom{\mathcal{S}}{N} \triangleq \{\mathcal{T} \subseteq \mathcal{S} : |\mathcal{T}| = N\}$. *For set* $\{a_n : n \in [N]\}$ *and* $\mathcal{N} \subseteq [N]$, $a_{\mathcal{N}}$ *denotes the set* $\{a_n : n \in \mathcal{N}\}$. *For the set of integers* $\{a_i : i \in [N]\}$ *we define* $\langle a_1, a_2, \ldots, a_N \rangle_C$ *to be the set* $\{b_i : b_i = C \text{ if } C \mid a_i, \text{ otherwise } b_i = a_i \mod C, i \in [N]\}$.

The following subsections briefly explain the PIR, coded caching and multi-user PIR problems. Firstly, we explain the single-user PIR problem of [2] and introduce the concept of private retrieval from distributed servers. Then, we introduce the coded caching problem [6] and its different variations, i.e., dedicated cache and multi-access coded caching problems. We provide motivation behind the cyclic-wraparound multi-access model in Section 1.3. Then, the combination of the dedicated cache-aided coded caching problem and the PIR problem as described in [7] is introduced in Section 1.4. Finally, the contribution of this paper, which considers a combination of PIR with a multi-access coded caching problem, is summarized in Section 1.5.

1.1. Private Information Retrieval

The protocol of Private Information Retrieval [2] allows for the retrieval of a specific file from a set of N files $\mathcal{W} = \{W_n\}_{n=1}^{N}$. These files are replicated across S non-colluding servers, with each file being of equal size. The objective of PIR is to retrieve the desired file, denoted as W_θ, without disclosing its identity to the servers. In other words, the user intends to conceal the index θ from the servers. To achieve this, the user generates S queries $\{Q_s^\theta\}_{s=1}^{S}$ and sends query Q_s^θ to server s. Upon receiving their respective queries, the servers generate answers based on the query received and the files they possess. Server s generates the answer $A_s^\theta(Q_s^\theta, \mathcal{W})$ and sends it back to the user. After receiving answers from all S servers, the user should be able to decode the desired file. The PIR protocol must satisfy privacy and correctness conditions, which are formally defined as follows:

Privacy condition:

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

Correctness condition:

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

The transmission cost of PIR is defined as

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

A PIR scheme is provided in [2], which incurs the minimum possible transmission cost, R^*_{PIR} is given as a function of the number of servers S and the number of files is denoted as N.

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

In the scheme provided in [2], every file has to be divided into S^N subfiles, and every server performs a transmission of size $(\frac{1}{S} + \frac{1}{S^2} + \ldots + \frac{1}{S^N})H(W_\theta)$.

Note: In the literature, the term rate is used for the transmission cost (e.g., [6]) as we use it here, whereas sometimes (e.g., [2]) the term rate is used for teh mean the inverse of the transmission cost as used by us. We use the term "transmission cost" instead of rate in this paper as in most of the coded caching literature [6,14,15].

1.2. Coded Caching

The authors in [6] propose a centralized coded caching system consisting of a server storing N independent files $W_0, \ldots, W_{N-1}$ of unit size and K users with a dedicated cache memory of size M files. However, in recent years, multi-access coded caching systems have been gaining attention, where C cache nodes exist, and each user can access several of them.

Coded caching systems work in two phases. In the *delivery phase*, which corresponds to low network traffic, the server fills the caches with the contents of the files. Then, in the *delivery phase*, all users wish to retrieve some files from the servers, increasing the network traffic. User k wishes to retrieve file W_{d_k} where $d_k \in [0 : N-1]$. Each user conveys to the server the index of the file they want. The server broadcasts coded transmissions $\mathbf{X}$ of size R_{CC} in the unit of files after receiving the user requests. The transmission $\mathbf{X}$ is a function of the files stored at the server and user demand. All users should be able to retrieve their chosen files using the caches they can access after receiving the coded transmission $\mathbf{X}$. The quantity R_{CC} is defined as the rate of the coded caching system, and it measures the size of the server's transmissions.

1.3. Multi-Access Coded Caching with Cyclic Wraparound Cache Access

Several approaches exist for users to access cache nodes in multi-access coded caching systems. In [15], multi-access schemes are derived from cross-resolvable designs, and the authors of [14] propose a system where each user can access L unique caches, resulting in $\binom{C}{L}$ users. This multi-access setting was further generalized in [16], showing that the rate achieved in [14] is optimal for certain cases. In this paper, we focus on a cyclic wraparound cache access approach where $C = K$ and each user accesses L neighboring cache nodes. This approach is reminiscent of circular wraparound networks, also known as ring networks, that have been extensively studied in the literature. For example, circular soft-handoff (SH) models in cellular networks [17] arrange nodes (base stations) in a circle, where users access only two nodes, their local node, and the node in the left neighboring cell. Another variant is the circular Wyner model, where nodes are arranged in a circle and users access three nodes (base stations), its local node, and nodes in two neighboring cells. Such settings were studied in [18], and Shannon-theoretic limits for a very simple cellular multiple-access system were obtained. In [19], the Wyner model was studied again, and upper and lower bounds on the per-user multiplexing gain of Wyner's circular soft-handoff model were presented. In [20], achievable rates were derived for the uplink channel of a cellular network with joint multicell processing. The rates were given in closed form for the classical Wyner model and the soft-handover model. There is extensive research on circular wraparound cache access in multi-access coded caching settings [8–13]. Like in cellular networks discussed above, this can occur when cache nodes are arranged in a circular manner and users access the L nearest cache nodes.

1.4. Dedicated Cache Aided MuPIR

In the dedicated cache setup described in [7], there is a collection of N files denoted as $\{W_n\}_{n\in[N]}$, which are replicated across S servers. The system involves K users, each equipped with a dedicated cache capable of storing M files. The users aim to retrieve their desired files from the servers. The system operates in two distinct phases.

In the *Placement Phase,* the cache of each user is populated with certain content. This cache content is determined based on the files stored across the servers and is independent of the future demands of the users. Subsequently, in the *Private Delivery Phase,* each user independently selects a file and seeks to privately retrieve their respective file from the servers. To achieve this, the users collaboratively generate S queries and transmit them to the servers. Upon receiving their respective queries, the servers respond with answers. The users should be able to decode their desired files using the transmitted answers and the content stored in their caches. In [7], an achievable scheme known as the *product design* is proposed. The product design results in a transmission cost denoted as R_{PD}, where

$$R_{PD}(M) = \min\{K - M, R'_{PD}(M)\} \text{ and} \tag{2}$$

$$R'_{PD}(M) = \frac{K(1-\frac{M}{N})}{\frac{KM}{N}+1} R^*_{PIR}(S,N) \tag{3}$$

whenever $M = tN/K$ for some integer $t \in [0:K]$. For other memory points, lower convex envelope of points $\{(M, R_{PD}(M)) : M = tN/K,\ t \in [0:K]\}$ is achieved by memory sharing.

Cache-aided multi-user PIR setups with multi-access caches are also considered in [21,22].

1.5. Our Contributions

This paper presents a PIR scheme that enables multiple users, aided by multi-access cache nodes, to privately retrieve data from distributed servers. The proposed scheme focuses on the multi-access setup with cyclic wraparound cache access where there are multiple non-colluding servers and all messages are replicated across these servers. The servers are connected with the users through noiseless broadcast links.

The contributions of this paper are as follows.

- The paper comprehensively describes the system model for the MACAMuPIR setup with cyclic wraparound cache access. It outlines the key components and mechanisms involved in the scheme.
- The paper presents an achievable scheme for the multi-access problem described above and characterizes its transmission cost.
- A comparison is made between the transmission costs of the multi-access setup and a dedicated cache setup proposed in previous work. The results show that the multi-access setup outperforms the dedicated cache setup.
- The paper includes proofs that validate the privacy guarantees and transmission costs mentioned in the scheme description. These proofs demonstrate the scheme's ability to preserve user privacy and ensure accurate retrieval of requested data.

1.6. Paper Organization

The rest of the paper is organized as follows:

- In Section 2, the problem statement is described, along with formal descriptions of transmission cost, privacy and correctness conditions.
- Then, in Section 3, the main results of the paper are summarized. The achieved rate is mentioned in this section.
- Section 4 has the scheme to achieve the transmission load mentioned in Section 3. We first explain the scheme using a concrete example in Section 4.1. Then, we extend the description to encompass general parameters in Section 4.2. We then specialize the

scheme to the context of cyclic wraparound cache access in Section 4.3. Then, proof of privacy and calculation of subpacketization level follows.

- After the specialized description of Section 4.3, we arrive at the critical observation that to calculate the rate, it is essential to characterize the number of $t + L$-sized subsets of $[K]$ that contain at least L consecutive integers, with wrapping around K allowed. Here, $t, K, L \in \mathbb{Z}$. This is calculated in Section 4.4 onward.
- Section 5 contains a discussion about the results and scope for future research, and Section 6 concludes the paper.

2. System Model: MACAMuPIR with Cyclic Wraparound Caches

The system consists of K users and N independent files, denoted as $\{W_n\}_{n\in[N]}$, which are replicated across $S \geq 2$ servers. Each file has a unit size. There are C cache nodes available, and each cache can store up to M files. Each user is connected to a unique set of $L \leq C$ cache nodes through links with infinite capacity. User k is connected to cache nodes indexed by $\mathcal{L}_k \in \binom{[C]}{L}$. The system follows a multi-access setup with cyclic wraparound cache access. In this setup, the number of users equals the number of cache nodes, i.e., $C = K$. Each user accesses L consecutive caches in a cyclic wraparound fashion. Specifically, user $k \in [K]$ accesses caches indexed by $\mathcal{L}_k = \langle k + l : l \in [0 : L - 1] \rangle_K$. We consider Figure 1; it is a multi-access system with cyclic wraparound access. In this system, we have $S = 2$ servers and $K = 4$ users. There are four cache nodes, and every user is accessing $L = 2$ cache nodes. The system operates in two phases described below.

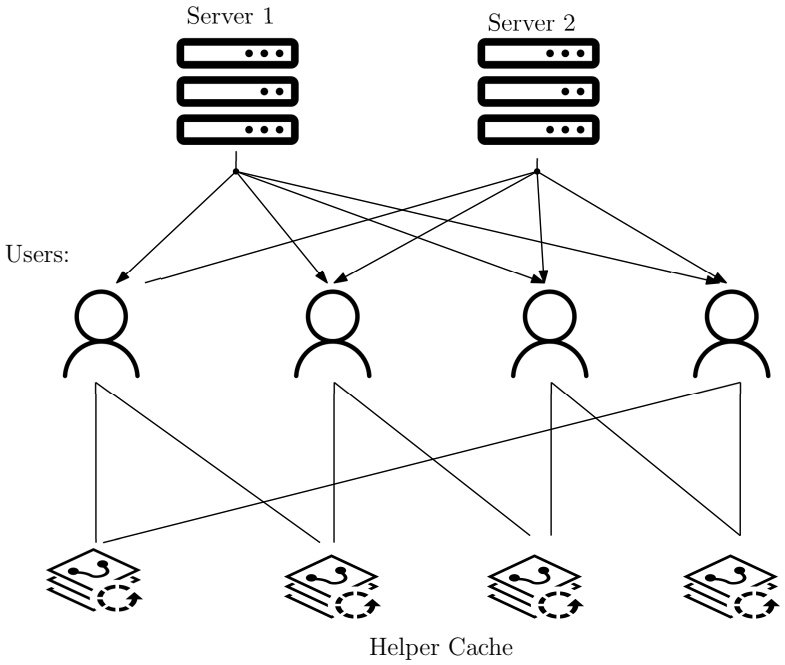

Figure 1. Multi-access coded caching setup with cyclic wraparound cache access with four users, four helper cache and two servers. Each user is accessing two adjacent helper caches.

Placement Phase: During this phase, all C cache nodes are populated. We let Z_c denote the content stored in cache $c \in [C]$. The content Z_c is determined based on the files $W_{[1:N]}$, and all servers possess knowledge of the content stored in each helper cache.

Private Delivery Phase: In this phase, each user aims to retrieve a specific file from the servers. User k selects $d_k \in [N]$ and desires to privately retrieve W_{d_k} from the servers. The demand vector is denoted as $\mathbf{d} = (d_1, d_2, \ldots, d_K)$. To retrieve their desired files from the servers while preserving privacy, users cooperatively generate S queries $Q_s^{\mathbf{d}}, s \in [S]$ based on their demands and the content stored in helper caches. These queries are designed in a way that they do not disclose the demand vector $\mathbf{d}$ to any of the servers. After generating the queries, each query $Q_s^{\mathbf{d}}$ is sent to server $s, \forall s \in [S]$. Upon receiving their respective

queries, each server $s, \forall s \in [S]$, broadcasts the answer $A_s^{\mathbf{d}}$ to the users. The answer is a function of the query $Q_s^{\mathbf{d}}$ and the files $W_{[1:N]}$. After receiving all S answers $A_{[S]}^{\mathbf{d}}$, each user $k, \forall k \in [K]$ decodes W_{d_k} using the caches accessible to that user.

To ensure the privacy of the user demands, the following condition must 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 have any information about user demands. And

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

known as the correctness condition, ensures that users experience no ambiguity concerning their desired file.

We define the transmission cost R as the amount of data that has to be transmitted by all the servers in order to satisfy the user demand.

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

Our goal is to design cache placement and private delivery schemes that satisfy privacy and correctness conditions and minimize the transmission cost.

3. Main Results: Achievable Rate and Comparison

In this section, we present the main result of the paper. For a given multi-access cache-aided MuPIR problem, we provide a scheme in Section 4 that can privately retrieve files from S non-colluding servers. For the multi-access cache aided MuPIR problem with cyclic wraparound cache access, the scheme incurs a transmission cost as described in Theorem 1. Then we compare the results of Theorem 1 with the dedicated cache-aided system of [7].

3.1. Achievable Rate

Before stating the transmission cost for cyclic wraparound cache access setup, we define the quantity $\text{cyc}(n, k, m)$ for integers $m \leq k < n$ as the number of k-sized subsets of n distinguishable elements arranged in a circle, such that there is at least one set of m consecutive elements amongst those k elements. An expression for $\text{cyc}(n, k, m)$ is given in Equation (4), the proof of which is given in Section 4.4.

$$\begin{aligned}\text{cyc}(n,k,m) = &\sum_{r=1}^{k}\left(\binom{n-k}{r}+\binom{n-k-1}{r-1}\right)\sum_{l=1}^{r}(-1)^{l-1}\binom{r}{l}\binom{k-l(m-1)-1}{r-1}\\ &+\sum_{r=3}^{k}\binom{n-k-1}{r-2}\Bigg(\sum_{l=2}^{m-1}(l-1)\sum_{j=1}^{r-2}(-1)^{j-1}\binom{r-2}{j}\binom{k-l-j(m-1)-1}{r-3}\\ &+\sum_{l=m}^{k}(l-1)\binom{k-l-1}{r-3}\Bigg)+k-1.\end{aligned} \tag{4}$$

Theorem 1. *For the cyclic wraparound multi-access coded caching setup, with S servers, N files, K helper caches and K users, where each user is accessing the L helper cache in a 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 revealing their demand to any of the servers, with*

$$R(t) = \min\{R_{PD}(M), R'(t)\}, \text{ where} \tag{5}$$

$$R'(t) = \frac{cyc(K, t+L, L)}{\binom{K}{t}} R^*_{PIR}(S, N). \tag{6}$$

Proof. In Section 4, we present a scheme that achieves $R'(t)$ as stated above for cyclic wraparound cache access setup. As users in the multi-access setup with cyclic wraparound cache access are accessing the caches, which the users of the dedicated cache setup are also accessing, the transmission cost of the multi-access setup are no higher than that of dedicated cache setup. For instance, if for some M, $R_{PD}(M) < R'(M)$, then the placement and delivery strategy of the product design can be employed. □

Theorem 1 characterizes a transmission cost in a multi-access setup where cache memory M is the 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.

3.2. Comparison with the Dedicated Cache Setup of [7]

We conduct a comparison between our scheme for cyclic wraparound multi-access systems and the product design proposed in [7]. In this comparison, we assume that the cache sizes and the number of users are identical in both settings. It is worth noting that the parameter $t = \frac{KM}{N}$ represents the number of times the entire set of N files can be replicated across the caches. For example, if $t = 2$, it implies that the cache nodes can store $2N$ units of data. Additionally, the total memory capacity of the system is tN units, which is equal to KM. It is important to mention that the transmission cost incurred by the product design is the same as the transmission cost presented in Theorem 1 for the special case where $L = 1$, indicating that each user only accesses one cache node.

To compare the transmission costs of both settings, we consider $K = 8$ users, $S = 2$ servers, and $N = 3$ files, and plot the transmission cost for various values of $t \in [8]$ and $L \in [7]$. The results are depicted in Figure 2. It can be observed that due to the access to a larger cache memory, the multi-access system outperforms the dedicated cache setup in terms of transmission cost.

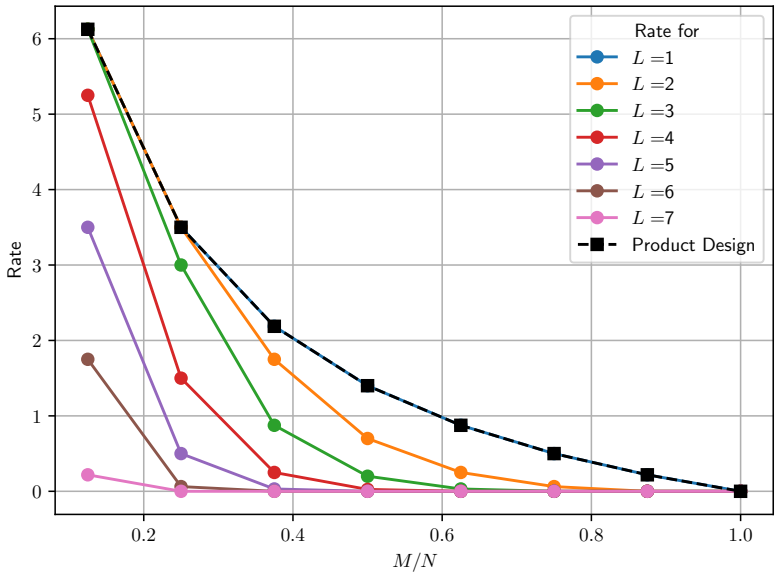

Figure 2. Comparison of transmission costs for dedicated cache (dotted lines) and multi-access (solid lines) with cyclic wraparound cache access. Here, we take $K = 8$ users and cache nodes.

4. Achievable Scheme: Proof of Theorem 1

4.1. Example

In this section, we present an achievable scheme using an example. Let us consider a cache-aided system with $N = 8$ files denoted as $W_1, W_2, \ldots, W_8$, $C = 8$ cache nodes, and $K = 8$ users. Each user has access to $L = 3$ cache nodes in a cyclic wraparound manner.

Since each user is connected to a unique set of three cache nodes, we can index each user with a subset of $[8]$ of size 3. For instance, the user connected to cache nodes indexed by six, seven, and eight can be denoted as $\{6,7,8\}$. Here is the list of all eight users:

$$\{1,2,3\},\{2,3,4\},\{3,4,5\},\{4,5,6\},\{5,6,7\},\{6,7,8\},\{1,7,8\},\{1,2,8\}.$$

Placement Phase: We let $t = \frac{CM}{N} = 1$. We divide each file into $\binom{8}{1} = 8$ subfiles, each indexed by integers in $[8]$.

$$W_n = \{W_{n,1}, W_{n,2}, W_{n,3}, W_{n,4}, W_{n,5}, W_{n,6}, W_{n,7}, W_{n,8}\}.$$

Then, we fill the cache nodes as follows:

$$\begin{aligned} &Z_1 = W_{[8],1} && Z_2 = W_{[8],2} && Z_3 = W_{[8],3} && Z_4 = W_{[8],4} \\ &Z_5 = W_{[8],5} && Z_6 = W_{[8],6} && Z_7 = W_{[8],7} && Z_8 = W_{[8],8}. \end{aligned}$$

Delivery Phase: In this phase, every user chooses one of the file indexes. We enumerate the demands of the users:

$$\begin{aligned} &d_{\{1,2,3\}} = 1 && d_{\{2,3,4\}} = 5 && d_{\{3,4,5\}} = 7 && d_{\{4,5,6\}} = 3 \\ &d_{\{5,6,7\}} = 1 && d_{\{6,7,8\}} = 2 && d_{\{1,7,8\}} = 8 && d_{\{1,2,8\}} = 4. \end{aligned}$$

For privately retrieving the files, users cooperatively generate queries as follows. For every $\mathcal{S} \in \binom{[8]}{4}$, such that $\mathcal{S}$ is the superset of at least one user index set, users generate sub-queries. For example, for $\mathcal{S} = \{3,4,5,6\}$, we have users $\{3,4,5\}$ and $\{4,5,6\}$ as a subset of $\{3,4,5,6\}$. Therefore, the users generate

$$\mathbf{Q}_s^{\mathbf{d},\{3,4,5,6\}} \triangleq \left\{ Q_{[2]}^{d_{\{3,4,5\}},\{3,4,5,6\}}, Q_{[2]}^{d_{\{4,5,6\}},\{3,4,5,6\}} \right\}$$

corresponding to $\{3,4,5,6\}$, where $Q_{[2]}^{d_{\{3,4,5\}},\{3,4,5,6\}}$ are the queries generated by the users in a single-user PIR setup if the demand is $d_{\{3,4,5\}} = 3$ and the set of files are $W_{[8],6}$, whereas there is no user for which $\{1,4,6,8\}$ is a superset; therefore, no queries can be generated corresponding to $\mathcal{S} = \{1,4,6,8\}$.

Then, for $\mathbf{Q}_s^{\mathbf{d},\{3,4,5,6\}}$, server s transmits

$$A_s^{d_{4,5,6}}\left(Q_s^{d_{4,5,6},\{3,4,5,6\}}, W_{[8],3}\right) + A_s^{d_{3,4,5}}\left(Q_s^{d_{3,4,5},\{3,4,5,6\}}, W_{[8],6}\right), \tag{7}$$

where $A_s^{d_{4,5,6}}\left(Q_s^{d_{4,5,6},\{3,4,5,6\}}, W_{[8],3}\right)$ is the answer of server s in a single-user PIR setup if the received query is $Q_s^{d_{\{4,5,6\}},\{3,4,5,6\}}$ and the set of files is $W_{[8],3}$.

Decoding

We consider user $\{4,5,6\}$ and subfiles $W_{3,3}$ and $W_{3,4}$. Subfile $W_{3,4}$ is available to the user from the cache node 4. Subfile $W_{3,3}$ has to be decoded from the transmissions. Consider the transmissions corresponding to $\mathcal{S} = \{3,4,5,6\}$ from (7):

User $\{4,5,6\}$ has access to subfiles $\{W_{[8],6}\}$, and therefore it can reconstruct

$$A_s^{d_{3,4,5}}\left(Q_s^{d_{3,4,5},\{3,4,5,6\}}, W_{[8],6}\right)$$

using the contents in Cache 6. After removing $A_s^{d_{3,4,5}}\left(Q_s^{d_{3,4,5},\{3,4,5,6\}}, W_{[8],6}\right)$ from the transmission corresponding to $\{3,4,5,6\}$, user $\{4,5,6\}$ obtains $A_s^{d_{4,5,6}}\left(Q_s^{d_{4,5,6},\{3,4,5,6\}}, W_{[8],3}\right)$, $\forall s \in \{1,2\}$. As these are the answers of a single-user PIR setup for demand $d_{\{4,5,6\}} = 3$ and files $W_{[8],3}$, user $\{4,5,6\}$ can decode $W_{3,3}$.

4.2. General Scheme: K Users, Each Connected to a Unique Arbitrary Set of L Caches

Consider N independent unit size files $\{W_n\}_{n\in[N]}$ replicated across the 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 the L cache, we index them with an L-sized subset of $[C]$. Specifically, user $\mathcal{K}$, where $\mathcal{K} \in \binom{[C]}{L}$, is the user connected to cache nodes indexed by $\mathcal{K}$. We let $\mathcal{U}$ be the set of all users where

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

Note that, for the special case of cyclic wraparound cache access, $C = K$ and $\mathcal{U} = \{\mathcal{L}_k : k \in [K]\}$ where $\mathcal{L}_k = \langle k + l : l \in [0 : L-1]\rangle_K$, $\forall k \in [K]$.

Placement Phase: We let $t = \frac{CM}{N}$ be an integer. Then, we 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, we 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 indexes. We let user $\mathcal{K}, \forall \mathcal{K} \in \mathcal{U}$ choose index $d_{\mathcal{K}} \in [N]$. User $\mathcal{K}$ then wishes to retrieve file $W_{d_{\mathcal{K}}}$ from the servers without reveling the index of the demanded file to the servers. We let $\mathbf{d} = (d_{\mathcal{K}})_{\mathcal{K}\in\mathcal{U}}$ be the demand vector. Users do not want the servers to obtain any information about the demand vector. For privately retrieving the files, the users cooperatively generate S queries $\mathbf{Q}_s^{\mathbf{d}}$ as follows. For every $\mathcal{S} \in \binom{[C]}{t+L}$, such that $\mathcal{S} \supset \mathcal{K}$ for at least one $\mathcal{K} \in \mathcal{U}$, the users generate sub-queries

$$\mathbf{Q}_s^{\mathbf{d},\mathcal{S}} = \left\{ Q_s^{d_{\mathcal{K}},\mathcal{S}} | \mathcal{K} \in \binom{\mathcal{S}}{L} \cap \mathcal{U} \right\} \tag{8}$$

where the sub-sub-query $Q_s^{d_{\mathcal{K}},\mathcal{S}}$ is the query sent to server s in a single-user PIR setup of [2] if the user demand is $d_{\mathcal{K}}$. We note that $\{Q_s^{d_{\mathcal{K}},\mathcal{S}}\}_{s\in[S]}$ for all $\mathcal{K}$ and for all $\mathcal{S}$ are generated independently. The query sent to server s is

$$\mathbf{Q}_s^{\mathbf{d}} = \left\{ \mathbf{Q}_s^{\mathbf{d},\mathcal{S}} | \mathcal{S} \in \binom{[C]}{t+L}, \mathcal{S} \supset \mathcal{K} \text{ for some } \mathcal{K} \in \mathcal{U} \right\}. \tag{9}$$

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

$$\bigoplus_{\mathcal{K}\in\binom{\mathcal{S}}{L}\cap\mathcal{U}} A_s^{d_{\mathcal{K}}}(Q_s^{d_{\mathcal{K}},\mathcal{S}}, W_{[N],\mathcal{S}\setminus\mathcal{K}}), \tag{10}$$

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 a single-user PIR setup if the received query is $Q_s^{d_{\mathcal{K}},\mathcal{S}}$ and the set of files is $\{W_{[N],\mathcal{S}\setminus\mathcal{K}}\}$.

Now, we proceed to show that all the users are able to decode their required file from these transmissions and the caches they have access to.

4.2.1. Decoding

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

$$\begin{aligned}&\bigoplus_{\mathcal{K}' \in \binom{\mathcal{K}\cup\mathcal{T}}{L} \cap \mathcal{U}} A_s^{d_{\mathcal{K}'}}(Q_s^{d_{\mathcal{K}}, \mathcal{K}\cup\mathcal{T}}, W_{[N],(\mathcal{K}\cup\mathcal{T})\setminus\mathcal{K}'}) \\ &= A_s^{d_{\mathcal{K}}}(Q_s^{d_{\mathcal{K}}, \mathcal{K}\cup\mathcal{T}}, W_{[N],\mathcal{T}}) \oplus \\ &\bigoplus_{\mathcal{K}' \in \binom{\mathcal{K}\cup\mathcal{T}}{L} \cap \mathcal{U}\setminus\mathcal{K}} A_s^{d_{\mathcal{K}'}}(Q_s^{d_{\mathcal{K}'}, \mathcal{K}\cup\mathcal{T}}, W_{[N],(\mathcal{K}\cup\mathcal{T})\setminus\mathcal{K}'}).\end{aligned} \tag{11}$$

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

4.2.2. Proof of Privacy

Now, we show that query $\mathbf{Q}_s^{\mathbf{d}}$ sent to server s is independent of the demand vector $\mathbf{d}$, $\forall s \in [S]$. For some $\mathcal{S}$, we consider $\mathbf{Q}_s^{\mathbf{d},\mathcal{S}}$ in (8). We show that $\mathbf{Q}_s^{\mathbf{d},\mathcal{S}}$ is independent of the demand vector. From the privacy of the single-user PIR scheme, the demand of user $\mathcal{K} \in \binom{\mathcal{S}}{L} \cap \mathcal{U}$, i.e., $d_{\mathcal{K}}$ is independent of sub-sub-query $Q_s^{d_{\mathcal{K}},\mathcal{S}}$. Also, other sub-sub-queries in $\mathbf{Q}_s^{\mathbf{d},\mathcal{S}}$ are similarly independent of $d_{\mathcal{K}}$ as they correspond to the users other than $\mathcal{K}$. This means that $\mathbf{Q}_s^{\mathbf{d},\mathcal{S}}$ is independent of the demands of the users in $\binom{\mathcal{S}}{L} \cap \mathcal{U}$. Moderover, all $Q_s^{d_{\mathcal{K}},\mathcal{S}}$ for any $\mathcal{K}$ and $\mathcal{S}$ are constructed independently, so these are also independent of the demands of users in $\mathcal{U} \setminus \binom{\mathcal{S}}{L}$. This shows that $\mathbf{Q}_s^{\mathbf{d},\mathcal{S}}$ is independent of the demand vector $\mathbf{d}$. Same analysis is true for any $\mathcal{S} \in \binom{[C]}{t+L} \cap \mathcal{U}$ where $\mathcal{S} \supset \mathcal{K}$ for at least one $\mathcal{K} \in \mathcal{U}$, which completes the proof of privacy for our scheme.

4.2.3. Subpacketization

As we can see, each file is divided into $\binom{C}{t}$ subfiles. According to the single-user PIR scheme, each of these subfiles has to be further divided into S^N sub-subfiles. Therefore, the subpacketization level is $\binom{C}{t} \times S^N$.

4.3. General Scheme: Cyclic Wraparound Cache Access

For cyclic wraparound cache access systems, we have $C = K$ and $\mathcal{U} = \{\mathcal{L}_k : k \in [K]\}$. Therefore, transmissions are performed only for those $\mathcal{S} \in \binom{[K]}{t+L}$ for which $\mathcal{L}_k \subset \mathcal{S}$ for at least one $k \in [K]$. This is the same as the number of $t+L$-sized subsets of $[K]$ that contain at least L consecutive integers, with wrapping around K allowed. As shown in Section 4.4, there are $\mathrm{cyc}(K, t+L, L)$ such subsets of $[K]$. Now, $\mathrm{cyc}(K, t+L, L)$ transmissions are performed by each of the S servers, and every transmission is of size $\left(\frac{1}{S} + \ldots + \frac{1}{S^N}\right)/\binom{K}{t}$ units; therefore, the transmission cost incurred is

$$R(t) = \frac{\mathrm{cyc}(K, t+L, L)}{\binom{K}{t}} \left(1 + \frac{1}{S} + \ldots + \frac{1}{S^{N-1}}\right).$$

Also, note that user k of the dedicated cache setup and that of the multi-access setup with a cyclic wraparound cache access are accessing cache node k. In a dedicated cache setup $\binom{K}{t+1}$, transmissions are required to satisfy user demands. Therefore, when $\mathrm{cyc}(K, t+L, L) > \binom{K}{t+1}$, we perform placement and transmissions as conducted for a dedicated cache setup. In this scenario, the transmission cost incurred in a multi-access setup is only as high as the transmission cost of a dedicated cache scenario with same cache sizes. For $t \in [0:K]$, the transmission cost of a multi-access setup would be

$$\min\left\{R_{PD}(\frac{tN}{K}), \frac{\mathrm{cyc}(K, t+L, L)}{\binom{K}{t}} R^*_{PIR}(S, N)\right\}.$$

We demonstrate our claim using an example for $K = 8$ caches and users, $S = 2$ servers, $N = 3$ files and $L = 2$. In Figure 3, we see that for smaller values of the t cyclic wraparound, cache access is incurring more transmission cost than the dedicated cache setup. For instance, when $M = 0.75$ or $t = 2$, $\text{cyc}(8, 4, 2) = 68$, transmissions are performed for cyclic wraparound cache access without memory sharing (and incurring transmission cost 4.25) compared to $\binom{8}{3} = 56$ transmissions in a dedicated cache setup (and incurring transmission cost 3.5). Therefore, when $t = 2$, transmissions corresponding to dedicated cache setup are performed. But when $M = 1.125$ or $t = 3$, a multi-access system with a cyclic wraparound cache access satisfy user demand with 56 transmissions (and an incurring transmission cost 1.75) compared to a dedicated cache setup which requires 70 transmissions (and sn incurring transmission cost 2.1875), and therefore transmissions, as described here, are performed. For cache memory M, $0.75 < M < 1.125$, memory sharing between these two schemes can incur transmission cost lower than either of these schemes.

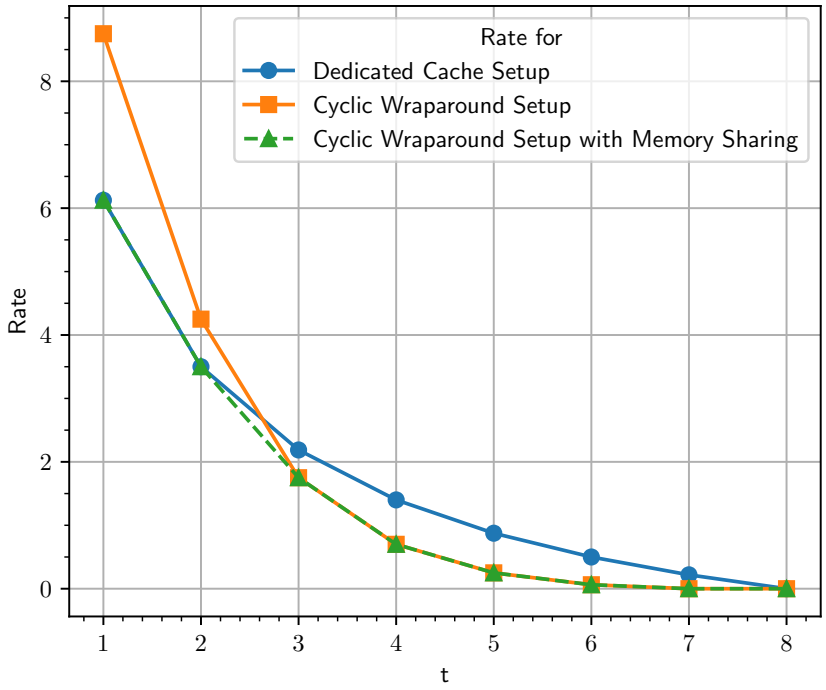

Figure 3. Transmission cost for $K = 8, L = 2, S = 2, N = 3$. Multi-access setup with cyclic wraparound cache access incur transmission cost only as high as dedicated cache setup with equal total memory in both systems.

4.4. Proving the Expression for cyc(n, k, m)

In this section, we show that the number of ways of choosing k integers from the set $[n]$, such that there is a subset of at least m consecutive integers, with cyclic wrapping around n allowed, is $cyc(n, k, m)$ as defined in (4).

First, for every $\mathcal{K} \in \binom{[n]}{k}$, we denote i_l as the length of l^{th} consecutive runs of integers inside $\mathcal{K}$ and o_l is the length of l^{th} consecutive run of integers outside $\mathcal{K}$. For instance, if $n = 10$ and $\mathcal{K} = \{1, 2, 4, 9, 10\}$, then $i_1 = 2$ corresponding to elements $\{1, 2\}$ in $\mathcal{K}$, $o_1 = 1$ corresponding to $\{3\}$ not in $\mathcal{K}$, $i_2 = 1$ corresponding to element $\{4\}$ in $\mathcal{K}$, $o_2 = 4$ corresponding to $\{5, 6, 7, 8\}$ not in $\mathcal{K}$ and $i_3 = 2$ corresponding to $\{9, 10\}$ in $\mathcal{K}$. Now, every $\mathcal{K} \in \binom{[n]}{k}$ can be uniquely determined by a sequence of positive integers consisting of i_l and o_l where every integer provides the length of consecutive runs of integers inside or outside $\mathcal{K}$, provided it is known if 1 is inside or outside $\mathcal{K}$. For example, with $n = 10$ and $k = 6$, if we are given the sequence of lengths of consecutive runs of the integers inside and outside $\mathcal{K}$ as $3, 2, 3, 2$ and it is known that $1 \in \mathcal{K}$, then we can uniquely figure out $\mathcal{K} = \{1, 2, 3, 6, 7, 8\}$.

Now, the set of all k-sized subsets $\mathcal{K}$ of $[n]$ with at least m cyclically consecutive integers can be partitioned into four disjoint sets as follows:

1. $1 \in \mathcal{K}$ and $n \notin \mathcal{K}$. This corresponds to sequences of the form $i_1, o_1, \ldots, i_r, o_r$ where $i_l, o_l \geq 1$ for all $l \in [r]$, $\sum_{l\in[r]} i_l = k$, $\sum_{l\in[r]} o_l = n-k$, $\exists l \in [r]$ such that $i_l \geq m$, $\forall r \in [k-m+1]$. We let the set of all such k-sized subsets be $\mathcal{K}_1$.
2. $1 \notin \mathcal{K}$ and $n \in \mathcal{K}$. This corresponds to sequences of the form $o_1, i_1, \ldots, o_r, i_r$ where $i_l, o_l \geq 1$ for all $l \in [r]$, $\sum_{l\in[r]} i_l = k$, $\sum_{l\in[r]} o_l = n-k$, $\exists l \in [r]$ such that $i_l \geq m$, $\forall r \in [k-m+1]$. We let the set of all such k-sized subsets be $\mathcal{K}_2$.
3. $1 \notin \mathcal{K}$ and $n \notin \mathcal{K}$. This corresponds to sequences of the form $o_1, i_1, \ldots, o_r, i_r, o_{r+1}$ where $i_l, o_l \geq 1$ for all $l \in [r+1]$, $\sum_{l\in[r]} i_l = k$, $\sum_{l\in[r+1]} o_l = n-k$ and $\exists l \in [r]$ such that $i_l \geq m$, $\forall r \in [k-m+1]$. The set of all such k-sized subsets is denoted by $\mathcal{K}_3$.
4. $1 \in \mathcal{K}$ and $n \in \mathcal{K}$. This corresponds to sequences of the form $i_1, o_1, \ldots, o_{r-1}, i_r$ where $i_l, o_l \geq 1$ for all $l \in [r]$, $\sum_{l\in[r]} i_l = k$, $\sum_{l\in[r]} o_l = n-k$ and $\exists l \in [2:r-1]$ such that $i_l \geq m$ or $x_1 + x_r \geq m$, $\forall r \in [k-m+1]$. We let the set of all such k-sized subsets be denoted by $\mathcal{K}_4$.

Now, we have $cyc(n,k,m) = |\mathcal{K}_1| + |\mathcal{K}_2| + |\mathcal{K}_3| + |\mathcal{K}_4|$. We proceed to calculate the size of these sets individually.

4.4.1. Calculation of $|\mathcal{K}_1|$

Sets in $\mathcal{K}_1$ correspond to the positive integer sequences of the form $i_1, o_1, \ldots, i_r, o_r$. Here, $\sum_{l\in[r]} i_l = k$ and $\sum_{l\in[r]} o_l = n-k$, and at least one $i_l \geq m$ and r takes all possible values in $[k-m+1]$.

We let I_j^r denote the set of tuples of r positive integers with the sum of integers equal to k and teh j^{th} integer greater than or equal to m, i.e.,

$$I_j^r = \{(i_1, i_2, \ldots, i_r) : \sum_{l\in[r]} i_l = k, i_l \geq 1, \forall l \in [r], i_j \geq m\}.$$

For a given r, $\cup_{j\in[r]} I_j^r$ is the set of all r length sequences, $(i_1, \ldots, i_r)$, of positive integers such that $\sum_{l\in[r]} i_l = k$. For all such sequences $i_1, \ldots, i_l$ there also exist $\binom{n-k-1}{r-1}$ sequences of positive integers $o_1, \ldots, o_r$ such that $\sum_{l\in[r]} o_l = n-k$. Therefore,

$$|\mathcal{K}_1| = \sum_{r\in[k-m+1]} \binom{n-k-1}{r-1} \Big| \bigcup_{j\in[r]} I_j^r \Big|.$$

From the inclusion–exclusion principle, we know that

$$\Big| \bigcup_{j\in[r]} I_j^r \Big| = \sum_{l=1}^{r} (-1)^{l-1} \sum_{1\leq j_1 < \cdots < j_l \leq r} \big| I_{j_1} \cap \cdots \cap I_{j_l} \big|,$$

where

$$\begin{aligned} \big| I_{j_1} \cap \cdots \cap I_{j_l} \big| &= |\{(i_1, \ldots, i_r) : \sum_{l\in[r]} i_l = k, i_l \geq 1, \forall l, i_{j_1}, \ldots, i_{j_l} \geq m\}| \\ &= |\{(i_1, \ldots, i_r) : \sum_{l\in[r]} i_l = k - l(m-1), i_l \geq 1, \forall l \in [r]\}| \\ &= \binom{k-l(m-1)-1}{r-1}, \end{aligned}$$

which implies

$$
\begin{aligned}
|\mathcal{K}_1| &= \sum_{r\in[k-m+1]} \binom{n-k-1}{r-1} \Big| \bigcup_{j\in[r]} I_j^r \Big| \\
&= \sum_{r\in[k-m+1]} \Big(\binom{n-k-1}{r-1} \times \sum_{l\in[r]} (-1)^{l-1} \sum_{1\le j_1<\cdots<j_l\le r} \binom{k-l(m-1)-1}{r-1} \Big) \\
&= \sum_{r\in[k-m+1]} \Big(\binom{n-k-1}{r-1} \times \sum_{l\in[r]} (-1)^{l-1} \binom{r}{l} \binom{k-l(m-1)-1}{r-1} \Big).
\end{aligned}
$$

4.4.2. Calculation of $|\mathcal{K}_2|$

By the definition of the set $\mathcal{K}_2$ and from the sequence of integers $o_1, i_1 \dots o_r, i_r$ corresponding to $\mathcal{K}_2$, it is clear that

$$|\mathcal{K}_2| = |\mathcal{K}_1|.$$

4.4.3. Calculation of $|\mathcal{K}_3|$

Here, we again see that we need a sequence of positive integers $i_1, \dots, i_r$ such that $\sum_{l\in[r]} = k$ and $\exists l \in [r]$ for which $i_l \ge m$. We have already calculated this quantity for $|\mathcal{K}_1|$, but for every such sequence of integers, there exist $\binom{n-k-1}{r}$ sequences $o_1, \dots, o_{r+1}$ of positive integers such that $\sum_{l\in[r+1]} o_l = n-k$. Therefore,

$$|\mathcal{K}_3| = \sum_{r=1}^{k-m+1} \binom{n-k-1}{r} \sum_{l\in[r]} (-1)^{l-1} \binom{r}{l} \binom{k-l(m-1)-1}{r-1}.$$

4.4.4. Calculation of $|\mathcal{K}_4|$

We consider all sequences of integers $i_1, o_1, \dots, o_{r-1}, i_r$ corresponding to $\mathcal{K}_4$ such that $i_l, o_l \ge 1$ for all $l \in [r]$, $\sum_{l\in[r]} i_l = k$, $\sum_{l\in[r]} o_l = n-k$ and $r \ge 2$ and $\exists l \in [r]$ such that $i_l \ge m$ OR $i_1 + i_r \ge m$. $\mathcal{K}_4$ can be partitioned into two disjoint subsets, $\mathcal{K}_{41}$ corresponding to sequences where $i_1 + i_r < m$ and $i_l \ge m$ for at lest one $l \in [2 : r-1]$ and $\mathcal{K}_{42}$ corresponding to sequences where $i_1 + i_r \ge m$. Again, $\mathcal{K}_4 = \mathcal{K}_{41} \cup \mathcal{K}_{42}$ and $\mathcal{K}_{41} \cap \mathcal{K}_{42} = \phi$. We proceed to calculate the cardinality of both these sets separately.

Calculation of $|\mathcal{K}_{41}|$

We consider the set of all r length positive integer sequences $i_1 \dots i_r$ such that $i_1 + i_r < m$ and $\sum_{l\in[r]} i_l = k$ and $i_l \ge m$ for some $l \in [2 : r-1]$. We note that, for such sequences, $r > 3$. The number of such sequences is

$$
\begin{aligned}
&\Big| \{(i_1, \dots, i_r) : \sum_{l\in[r]} i_l = k, i_l \ge 1, i_1 + i_r < m, \exists l \text{ s.t. } i_l \ge m\} \Big| \\
&= \sum_{s=2}^{m-1} (s-1) \Big| \{(i_2 \dots i_{r-1}) : \sum_{l=2}^{r-1} i_l = k-s, i_l \ge 1, \exists l \text{ s.t. } i_l \ge m\} \Big| \\
&= \sum_{s=2}^{m-1} (s-1) \sum_{j=1}^{r-2} (-1)^{j-1} \binom{r-2}{j} \binom{k-s-j(m-1)-1}{r-3}.
\end{aligned}
$$

For every such r length sequence, there exist $\binom{n-k-1}{r-2}$ positive integer sequences $o_1 \dots o_{r-1}$ such that $\sum_{l\in[r-1]} o_l = n-k$, and we obtain

$$|\mathcal{K}_{41}| = \sum_{r=3}^{k-m+1} \Big(\binom{n-k-1}{r-2} \sum_{s=2}^{m-1} (s-1) \times \sum_{j=1}^{r-2} (-1)^{j-1} \binom{r-2}{j} \binom{k-s-j(m-1)-1}{r-3} \Big).$$

Calculation of $\left|\mathcal{K}_{42}\right|$

We consider the set of all $r > 2$ length positive integer sequences $i_1 \dots i_r$ such that $i_1 + i_r \geq m$ and $\sum_{l\in[r]} i_l = k$. The number of such sequences is

$$\begin{aligned}
&\left|\{(i_1 \dots i_r) : \sum_{l\in[r]} i_l = k, i_1 + i_r \geq m, i_l \geq 1\}\right| \\
&= \sum_{s=m}^{k-(r-2)} (s-1)\left|\{(i_2 \dots i_{r-1}) : \sum_{l\in[2:r-1]} i_l = k-s, i_l \geq 1\}\right| \\
&= \sum_{s=m}^{k-(r-2)} (s-1)\binom{k-s-1}{r-3}.
\end{aligned}$$

For every such r length sequence, there exist $\binom{n-k-1}{r-2}$ positive integer sequences $o_1 \dots o_{r-1}$ such that $\sum_{l\in[r-1]} = n-k$, and when $r = 2$, there are $k-1$ possible pairs of positive integers i_2, i_2 which provides $i_1 + i_2 = k$, leading to

$$|\mathcal{K}_{42}| = \sum_{r=2}^{k-m+1} \binom{n-k-1}{r-2} \sum_{s=m}^{k-(r-2)} (s-1)\binom{k-s-1}{r-3} + k - 1.$$

Finally, we have

$$\begin{aligned}
&cyc(n,k,m) = |\mathcal{K}_1| + |\mathcal{K}_2| + |\mathcal{K}_3| + |\mathcal{K}_{41}| + |\mathcal{K}_{42}| \\
&= \sum_{r=1}^{k-m+1} \binom{n-k-1}{r-1} \sum_{l\in[r]} (-1)^{l-1}\binom{r}{l}\binom{k-l(m-1)-1}{r-1} \\
&+ \sum_{r=1}^{k-m+1} \binom{n-k-1}{r-1} \sum_{l\in[r]} (-1)^{l-1}\binom{r}{l}\binom{k-l(m-1)-1}{r-1} \\
&+ \sum_{r=1}^{k-m+1} \binom{n-k-1}{r} \sum_{l\in[r]} (-1)^{l-1}\binom{r}{l}\binom{k-l(m-1)-1}{r-1} \\
&+ \sum_{r=3}^{k-m+1} \binom{n-k-1}{r-2} \sum_{s=2}^{m-1}\Big((s-1) \\
&\times \sum_{j\in[r-2]} (-1)^{j-1}\binom{r-2}{j}\binom{k-s-j(m-1)-1}{r-3}\Big) \\
&+ \sum_{r=3}^{k-m+1} \binom{n-k-1}{r-2} \sum_{s=m}^{k-(r-2)} (s-1)\binom{k-s-1}{r-3} + k - 1.
\end{aligned}$$

Defining $\binom{a}{b} = 0$ if $a < b$ or if $a < 0$ or $b < 0$, the expression above can be simplified to (4).

5. Discussion

In this paper, we proposed an efficient and privacy-preserving scheme for multi-user retrieval scenarios. By leveraging the benefits of multi-access setups with cyclic wraparound cache access, we demonstrated improved transmission costs compared to the dedicated cache setup. We conducted a comprehensive comparison with prior works that utilize dedicated cache systems. Our results demonstrate the superior performance of our proposed scheme.

Moreover, the placement and delivery schemes designed for dedicated cache-aided MuPIR scenarios are applicable in our MAC-MuPIR scenarios with cyclic wraparound cache access. This adaptability leads to consistently lower rates in our scheme compared to the product design. Hence, we achieved even lower rates by utilizing memory sharing between our setup and the dedicated cache-aided setup. For instance, if we consider Figure 3, specifically for points $t = 1, 2.5$ and 4. At $t = 1$, the dedicated cache setup of [7]

achieves a lower rate than the scheme provided in Section 4.3, so placement and delivery, as described in [7], are performed, which still work in our setting. At $t = 4$, the scheme provided in Section 4.3 has a lower rate, so placement and delivery are performed as described in the section mentioned above. At $t = 2.5$, we can perform memory sharing between dedicated cache and multi-access cache schemes and achieve a rate lower than both of the schemes (dotted line in Figure 3, referring to scheme with memory sharing below the lines corresponding to dedicated cache scheme and scheme mentioned in Section 4.3).

However, it is important to acknowledge some limitations of our study. Firstly, we focused on noiseless broadcast links, which may not reflect real-world scenarios where channel impairments exist. Future research could investigate the impact of channel conditions on the performance of the proposed scheme. Additionally, we assumed non-colluding servers and replicated messages across the servers. Exploring the scheme's resilience in the presence of adversarial behaviors or server failures could be an interesting direction for further investigation.

6. Conclusions

In this study, we introduced a PIR scheme that enables multiple users to securely retrieve data from distributed servers using a multi-access setup with cyclic wraparound cache access. We described the system model, formally defined the privacy and correctness constraints, and presented the transmission cost associated with our proposed scheme.

Our findings indicate that the multi-access setup with cyclic wraparound cache access offers significant advantages over the dedicated cache setup. By comparing the transmission costs of both setups, we demonstrated that the multi-access setup outperforms the dedicated cache setup, making it a more efficient and reliable approach for multi-user PIR scenarios. For instance, in Figure 2, for caching ratio $M/N = 0.25$, we see that for $L = 2$ the cyclic wraparound system and the dedicated cache system both achieve a rate of three. But the rate decreases as cache access degree L increases and users have access to more caches. For $L = 3, 4, 5, 6$ and 7, the rate of our scheme is $3, 1.5, 0.5, 0.0625$ and 0, respectively. More than a twofold improvement in download cost is shown compared to that of the dedicated cache setup from $L = 4$ onward, i.e., accessing half of the available caches.

Furthermore, our scheme provides strong privacy guarantees, ensuring that users can retrieve data without revealing their individual retrieval patterns or compromising the privacy of the data. The proofs presented in Section 4.2.2 validate the privacy and transmission costs associated with our scheme, reinforcing its effectiveness and security.

Author Contributions: Conceptualization, K.V. and B.S.R.; methodology, B.S.R.; software, K.V.; validation, B.S.R.; formal analysis, K.V.; investigation, K.V. and B.S.R.; data curation, K.V.; writing—original draft preparation, K.V.; writing—review and editing, B.S.R.; visualization, K.V. and B.S.R.; supervision, B.S.R.; project administration, B.S.R.; funding acquisition, B.S.R. All authors have read and agreed to the published version of the manuscript.

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

Institutional Review Board Statement: Not applicable.

Data Availability Statement: Data sharing not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Chor, B.; Goldreich, O.; Kushilevitz, E.; Sudan, M. Private information retrieval. In Proceedings of the IEEE 36th Annual Foundations of Computer Science, Milwaukee, WI, USA, 23–25 October 1995; pp. 41–50. [CrossRef]
2. Sun, H.; Jafar, S.A. The Capacity of Private Information Retrieval. *IEEE Trans. Inf. Theory* **2017**, *63*, 4075–4088. [CrossRef]
3. Sun, H.; Jafar, S.A. The Capacity of Robust Private Information Retrieval With Colluding Databases. *IEEE Trans. Inf. Theory* **2018**, *64*, 2361–2370. [CrossRef]

4. Lin, H.Y.; Kumar, S.; Rosnes, E.; Amat, A.G.i.; Yaakobi, E. Weakly-Private Information Retrieval. In Proceedings of the 2019 IEEE International Symposium on Information Theory (ISIT), Mutualité, France, 7–12 July 2019; pp. 1257–1261. [CrossRef]
5. Chen, Z.; Wang, Z.; Jafar, S.A. The Capacity of T-Private Information Retrieval with Private Side Information. *IEEE Trans. Inf. Theory* **2020**, *66*, 4761–4773. [CrossRef]
6. Maddah-Ali, M.A.; Niesen, U. Fundamental Limits of Caching. *IEEE Trans. Inf. Theory* **2014**, *60*, 2856–2867. [CrossRef]
7. Zhang, X.; Wan, K.; Sun, H.; Ji, M.; Caire, G. On the Fundamental Limits of Cache-Aided Multiuser Private Information Retrieval. *IEEE Trans. Commun.* **2021**, *69*, 5828–5842. [CrossRef]
8. Hachem, J.; Karamchandani, N.; Diggavi, S.N. Coded Caching for Multi-level Popularity and Access. *IEEE Trans. Inf. Theory* **2017**, *63*, 3108–3141. [CrossRef]
9. Reddy, K.S.; Karamchandani, N. Rate-Memory Trade-off for Multi-Access Coded Caching With Uncoded Placement. *IEEE Trans. Commun.* **2020**, *68*, 3261–3274. [CrossRef]
10. Trinadh, P.; Dutta, M.; Thomas, A.; Rajan, B.S. Decentralized Multi-access Coded Caching with Uncoded Prefetching. In Proceedings of the 2021 IEEE Information Theory Workshop (ITW), Virtual, 17–21 October 2021; pp. 1–6. [CrossRef]
11. Cheng, M.; Wan, K.; Liang, D.; Zhang, M.; Caire, G. A Novel Transformation Approach of Shared-Link Coded Caching Schemes for Multiaccess Networks. *IEEE Trans. Commun.* **2021**, *69*, 7376–7389. [CrossRef]
12. Sasi; Shanuja; Rajan, B.S. An improved multi-access coded caching with uncoded placement. *arXiv* **2020**, arXiv:2009.05377.
13. Serbetci, B.; Parrinello, E.; Elia, P. Multi-access coded caching: Gains beyond cache-redundancy. In Proceedings of the 2019 IEEE Information Theory Workshop (ITW), Visby, Sweden, 25–28 August 2019; pp. 1–5. [CrossRef]
14. Muralidhar, P.N.; Katyal, D.; Rajan, B.S. Maddah-Ali-Niesen Scheme for Multi-access Coded Caching. In Proceedings of the IEEE Information Theory Workshop, (ITW2021), Kanazawa, Japan, 17–21 October 2021.
15. Katyal, D.; Muralidhar, P.N.; Rajan, B.S. Multi-access Coded Caching Schemes From Cross Resolvable Designs. *IEEE Trans. Inf. Theory* **2021**, *69*, 2997–3010. [CrossRef]
16. Brunero, F.; Elia, P. Fundamental Limits of Combinatorial Multi-access Caching. *IEEE Trans. Inf. Theory* **2022**, *69*, 1037–1056. [CrossRef]
17. Somekh, O.; Zaidel, B.M.; Shamai, S. Spectral Efficiency of Joint Multiple Cell-Site Processors for Randomly Spread DS-CDMA Systems. *IEEE Trans. Inf. Theory* **2007**, *53*, 2625–2637. [CrossRef]
18. Wyner, A.D. Shannon-theoretic approach to a Gaussian cellular multiple-access channel. *IEEE Trans. Inf. Theory* **1994**, *40*, 1713–1727. [CrossRef]
19. Wigger, M.; Timo, R.; Shamai, S. Complete interference mitigation through receiver-caching in Wyner's networks. In Proceedings of the 2016 IEEE Information Theory Workshop (ITW), Cambridge, UK, 11–14 September 2016; pp. 335–339. [CrossRef]
20. Sanderovich, A.; Somekh, O.; Poor, H.V.; Shamai, S. Uplink Macro Diversity of Limited Backhaul Cellular Network. *IEEE Trans. Inf. Theory* **2009**, *55*, 3457–3478. [CrossRef]
21. Vaidya, K.; Rajan, B.S. Multi-Access Cache-Aided Multi-User Private Information Retrieval. *arXiv* **2022**, arXiv:2201.11481.
22. Vaidya, K.; Rajan, B.S. Cache-Aided Multi-Access Multi-User Private Information Retrieval. In Proceedings of the 2022 20th International Symposium on Modeling and Optimization in Mobile, Ad hoc, and Wireless Networks (WiOpt), Torino, Italy, 19–23 September 2022; pp. 246–253. [CrossRef]