Maddah-Ali-Niesen Scheme for Multi-access Coded Caching

Pooja Nayak Muralidhar, Digvijay Katyal and B. Sundar Rajan
Department of Electrical Communication Engineering, Indian Institute of Science, Bengaluru 560012, KA, India
E-mail: {poojam,digvijayk,bsrajan}@iisc.ac.in

Abstract—The well known Maddah-Ali-Niesen (MAN) coded caching scheme for users with dedicated cache is extended for use in multi-access coded cache scheme where the number of users need not be same as the number of caches in the system. The well known MAN scheme is recoverable as a special case of the multi-access system considered. The performance of this scheme is compared with the existing works on multi-access coded caching. To be able to compare the performance of different multi-access schemes with different number of users for the same number of caches, the terminology of per user rate (rate divided by the number of users) introduced in [17] is used.

I. INTRODUCTION

Due to the increasing number of multimedia applications like video on demand, tremendous growth in the consumption of data has been observed in recent years. The seminal work of [11] showed that jointly designing content placement and delivery, also known as coded caching significantly improves content delivery rate requirements. However in reality, due to practical constraints the subpacketization levels in [11] is not feasible due to its exponentially increasing nature, with respect to the increasing number of users. The quest for coded caching schemes with practical subpacketization levels started, and a wide variety of schemes using different constructions of placement delivery arrays [2], line graphs of bipartite graphs [3], linear block codes [4], block designs [5] etc. were found.

Parallely, this technique of jointly designing placement and delivery were explored in other type of networks like device to device networks (D2D) [6], Combination networks [7], networks with shared cache [8], [9], [10] etc. The case of networks with shared caches is particularly interesting since in these types of networks users can share the caches and hence in a practical perspective it helps in efficient utilization of memory. Coded caching in networks where demands are non uniform i.e., files with different popularities [11], [12] has also been well studied. However we will be limiting ourselves to the cases where all the files are equipopular and demands are distinct.

An interesting and practical type of network scenario is a multi-access network where multiple users can access the same cache and multiple caches can be accessed by the same user [12], [15], [16], [17], [18], [19], [20]. The scenario considered in [12], [15], [16], [18], [19] and [20] has $K$ users and $K$ caches and a "sliding window" approach, where user $k$ accesses caches $k, k+1, \ldots, k+r-1$ for some $r \in 1, 2, \ldots, K$, using a cyclic wraparound to preserve symmetry is used. Here, $r$ is called cache access degree i.e. the number of caches a user has access to.

In the following subsection we provide a brief survey of the schemes that are known in the literature of multi-access networks.

Known schemes in multi-access

1) Hachem-Karamchandani-Diggavi (HKD) Scheme [12]: Multi-access setups were introduced in the work of [12] where caching and delivery in a decentralized setting with multi-access and multi-level popularities under consideration. The authors give rate memory trade offs for a multi-level access model (content is divided into discrete levels based on popularity and users are required to connect to a certain number of access points based on the popularity of the file they have requested) with multi user setups (user can access multiple caches). In the case of a multi-user, multi-access model with a single-level caching system, with $N$ files, $K$ caches and $K$ users grouped into $U$ users in a group with access degree $r$, such that $N \geq KU$ and $r$ divides $K$, and a cache memory of $M \in [0, \frac{K}{r}]$, in the decentralized assumption, considered in the paper, the achievable rate is given by

$$R = U \min \left\{ \frac{N}{M}, K \right\} \left( 1 - \frac{rM}{N} \right).$$

Also when $r$ does not divide $K$, four times the above expression can be achieved and

$$R = 0 \text{ if } M > \frac{N}{r}.$$

In [12], the scheme proposed is decentralized. However the scheme can be extended to get a centralized scheme (also indicated in [15]) with the rate achieved given by

$$R = \frac{K - K r M}{1 + K r M}.$$

Henceforth we refer to this scheme as the HKD scheme.

2) Reddy-Karamchandani (RK) Scheme [15]: The scheme proposed in [15] supports a multi-access setup with $K$ users and $K$ caches with each user connected to $r$ consecutive caches in a cyclic manner. The rate for this scheme for
A generic lower bound on the optimal rate for any such multi-access setup with $r \geq \frac{K}{2}$, under the restriction of uncoded placement is derived as

$$R_{lb}(M) = \begin{cases} 
K - \frac{(K-r)(K-r+1)}{2K} \frac{MK}{N}, & \text{if } 0 \leq M \leq \frac{N}{K} \\
\frac{(K-r)(K-r+1)}{2K} (2 - \frac{MK}{N}), & \text{if } \frac{N}{K} \leq M \leq \frac{2N}{K} \\
0, & \text{if } M \geq \frac{2N}{K}.
\end{cases}$$

The rate achieved by this scheme is proved to be order optimal with the multiplicative gap between the achievable rate and lower bound, at most 2 for $r \geq K/2$. The scheme is also found to be optimal for the special cases, $r = K - 1, r = K - 2, r = K - 3$ with $K$ even and $r = K - \frac{K}{4} + 1$ for some positive integer $s$. Hereafter, we refer to this scheme as the RK scheme.

3. Serbetci-Parinello-Elia (SPE) Scheme [16]: The work in [16] also deals with the same problem setup as in the previous cases and provides two new schemes which can serve, on average, more than $K\gamma + 1$ users at a time and for the special case of $r = \frac{K}{2}$, the achieved gain is proved to be optimal under uncoded cache placement where $\gamma = \frac{1}{\gcd(K, t)}$, $\gamma \in \left\{\frac{1}{K}, \frac{1}{2}, \frac{1}{3}, \ldots, 1\right\}$. The general scheme is proposed for the case $K\gamma = 2$. The subpacketization for this scheme is given by $F = \frac{K(K-2r+2)}{2K}$ and it can be noted that the numerator should be divisible by 4 and $r < \frac{(K+2)}{2}$, for the number of subpackets to be a positive integer greater than 0. We refer to this scheme as the SPE scheme.

4. Scheme using Cross Resolvable Designs (CRD) [17]: In [17] the authors develop a multi-access coded caching scheme from a specific type of resolvable designs called cross resolvable designs. The number of users supported in this scheme is higher than the other existing schemes for the same number of caches and for practically realizable subpacketization levels. To compare the performance with other schemes, the authors introduce the notion of per user rate or rate per user obtained by normalizing the rate $R$ with the number of users $K$ supported, i.e., rate per user is $\frac{R}{K}$. We refer to this scheme as the CRD scheme.

5. Cheng-Liang-Wan-Zhang-Caire (CLWZC) Scheme [18]: The work of [18] proposes a novel transformation approach to extend the Maddah Ali Niesen scheme to the multiaccess caching systems, such that the load expression of the scheme by [12] remains achievable even when $r$ does not divide $K$. This work considers only the multi-access setup with cyclic wrapper around where each user has access to $r$ neighboring cache-nodes with a cyclic wrapper around setup. The rate expression for this scheme is same as that for the centralized HKD scheme. The subpacketization required is $K(K - (r-1))$, where $t = \frac{KM}{N}$.

A. Sasi Rajan Scheme 1 [19]

In [19], multi-access schemes based on cyclic setup with some improvement over some of the cases in RK scheme and CLWZC scheme has been proposed. The scheme proposed exists for the cases when normalized capacity $\gamma$, where $\gamma \in \left\{\frac{1}{K}, \frac{1}{t} : \gcd(t, K) = 1, t \in [1, K]\right\}$. The rates achieved here, are given below.

- $\mathcal{R}_{SR1}(\gamma) = \frac{1}{\gamma}$, if $(K - tr) = 1$.
- If $(K - tr)$ is even, then $\mathcal{R}_{SR1}(\gamma) = 2\sum_{i=1}^{\frac{K-tr}{2}} \frac{1}{1+2\gamma}$.
- If $(K - tr) > 1$ is odd, then $\mathcal{R}_{SR1}(\gamma) = \left(\frac{1}{\gamma - (tr+1)}\right) + \sum_{i=1}^{\frac{K-tr}{2}} \frac{1}{1+2\gamma}$.

The subpacketization achieved here is at most $K^2$. The scheme outperforms the RK scheme in all cases other than the ones achieving optimality. The scheme proposed performs better than the CLWZC scheme when $r \geq \left(1 - \frac{1}{2}\right) \frac{K}{2}$. This scheme is a generalisation of one of the cases considered in [16], i.e., when $r = \frac{K}{4\gamma}$. We will refer to this scheme as the SR scheme 1.

B. Sasi Rajan Scheme 2 [20]

In [20], multi-access schemes based on placement delivery arrays have been proposed in the work of [20], where subpacketization level varies only linearly with the number of users. The rate achieved by this scheme is

$$R_{SR2} = \frac{(K - tr)(K - tr + t)}{2K}$$

where $\gamma \in \left\{\frac{1}{K}, \frac{1}{t}: t\{K, (K - tr + t)|K, t \in [1, K]\right\}$ and subpacketization $F = K$. The scheme of [20] is found to be better than the CLWZC scheme for $r \geq \frac{K(t+1)}{t(t+1)} + 1$. This scheme also has performance better than or equal to that of RK scheme. We will refer to this scheme as the SR scheme 2.

C. Our Contributions

In this work a coded caching scheme for multi-access networks is proposed with the number of users much larger than the number of caches. The proposed scheme is extension of the MAN scheme and the MAN scheme can be obtained as a special case. The performance of this scheme is compared with the existing works. To be able to compare the performance of different multi-access schemes with different number of users for the same number of caches, the terminology of per user rate $\frac{R}{K}$ introduced in [17] is used.

Notations: The set $\{1, 2, \ldots, n\}$ is denoted as $[n]$, $|\mathcal{X}|$ denotes cardinality of the set $\mathcal{X}$.

II. THE PROPOSED SCHEME

Our problem set up is as above in Fig. [1]. Let $\mathcal{K}$ denote the set of $K$ users in the network connected via an error free shared link to a server $\mathcal{S}$ storing $N$ files ($N \geq K$) denoted as $W_1, W_2, W_3, \ldots, W_N$ each of unit size. Each user can access a unique set of $r$ caches (cache access degree) out of $C$ caches, each capable of storing $M$ files. $Z_k$ denotes the content in
It is seen that after the placement, the size of the content stored places the content given by \( Z \). Let \( C \) denote the capacity link to the caches that it is connected to. Let \( U \subset C \); \( |U| = r \) denote the demand of the user connected to the set of caches represented by \( U \). Hence, the maximum number of users \( K \) in this scheme is \( \binom{C}{r} \). Our scheme works for any number of users as long as a user is associated with a unique subset of \( r \) caches, i.e., no two users are accessing the same set of \( r \) caches. For concreteness we assume that \( K = \binom{C}{r} \). The scheme works in two phases described below.

**Placement Phase:** Let \( t = \frac{CM}{N} \) be an integer. The server divides each file \( W_i \) into \( \binom{C}{r} \) subfiles and in the \( k^{th} \) cache places the content given by

\[
Z_k = \{ W_{i,T} : k \in T, \ T \subset C, \ |T| = t, \ \forall i \in [N] \}.
\]

It is seen that after the placement, the size of the content stored in the cache is equal to \( \binom{C}{r} N = \frac{t}{r} C = M \).

Note that a user may find the same subfiles of a file in more than one cache it has access to. Let \( M' \) denote the size of the files that a user has access to. We have

\[
\frac{M'}{N} = \sum_{i=1}^{r} \frac{|Z_i|}{\binom{C}{r}} - \sum_{1 \leq i_1 < i_2 \leq r} \frac{|Z_{i_1} \cap Z_{i_2}|}{\binom{C}{r}} + \cdots + \sum_{1 \leq i_1 < \cdots < i_s \leq r} \frac{|Z_{i_1} \cap \cdots \cap Z_{i_s}|}{\binom{C}{r}} + \cdots + \frac{|Z_1 \cap \cdots \cap Z_r|}{\binom{C}{r}} - \binom{t}{C} + 1
\]

where \( Z_{i_k}, i_k \in [r] \) are distinct \( r \) caches. The above expression simplifies to

\[
\frac{M'}{N} = \binom{t}{1} \binom{C-1}{r-1} + \binom{t}{2} \binom{C-2}{r-2} + \cdots + (-1)^{(s+1)} \binom{t}{s} \binom{C-s}{r-s} + \cdots + (-1)^{(r+1)} \binom{t}{r} \binom{C-r}{r}
\]

**Remark 1:** Note that the placement is same as the placement of Maddah Ali Niesen scheme in shared link networks \( \Pi \) with the difference that in \( \Pi \), each user is equipped with a dedicated cache.

**Delivery Phase:** For each such subset \( S \subset C \) of cardinality \( |S| = t + r \), the server transmits

\[
\mathcal{Y}_S = \bigoplus_{U \subset S, |U| = r} W_{d,U}, \mathcal{S}_{\setminus U}
\]

Now we prove that with the above delivery scheme every user will be able to decode the file it wants to.

**Proof:** First, we note that when \( U \cap \mathcal{T} = \emptyset \), where \( \mathcal{T} \subset C, \ |\mathcal{T}| = t \), then the user \( U \) does not have access to any cache \( c \) where, \( c \in \mathcal{T} \) and \( c \notin U \). A user \( U \) has access to a subfile indexed by \( \mathcal{T} \) iff \( U \cap \mathcal{T} \neq \emptyset \). This is because, when \( U \cap \mathcal{T} \neq \emptyset \), there exists a cache \( c \in U \cap \mathcal{T} \), through which \( U \) can access the subfile indexed by \( \mathcal{T} \).

Consider the delivery algorithm mentioned above. We now argue that each user can successfully recover its requested message. Consider the transmission by the server

\[
\bigoplus_{U \subset S, |U| = r} W_{d,U}, \mathcal{S}_{\setminus U}
\]

corresponding to the subset of caches, \( S \subset C \) with \( |S| = t + r \). Consider a user \( U \subset S; \ |U| = r \). The user \( U \) already has access to the subfiles \( W_{d,U}, \mathcal{S}_{\setminus U} \) for any other user indexed by \( \mathcal{V} \subset \{S : \mathcal{V} \notin U\}; \ |\mathcal{V}| = r \) since, \( \{S \setminus \mathcal{V}\} \cap \mathcal{U} = \emptyset \). Hence it can retrieve the subfile \( W_{d,U}, \mathcal{S}_{\setminus (U \setminus S)} \) from the transmission corresponding to \( S \). Likewise, from all such transmissions corresponding to \( S \), with \( U \subset S \), user \( U \) gets the missing subfiles of \( W_{d,U} \) sent over the shared link. Since this is true for every such subset \( S \), any user \( U \subset C; \ |U| = r \) can recover all missing subfiles.

**Theorem 1:** For every integer \( t = \frac{CM}{N} \), the above placement and delivery results in a multi-access scheme with \( C \) caches, access degree \( r \), \( \binom{C}{r} \) users, subpacketization \( F = \binom{t}{r} \), coding gain \( g = \frac{r}{C} \), and rate \( R = \frac{C}{r} \binom{r}{t} \).

**Proof:** According to the proposed scheme, the size of each subpacket is \( \frac{t}{r} \). The total number of transmissions is the total number of choices of \( S = \binom{C}{t+r} \). The coding gain, defined as the total number of users benefited from each transmission is thus, the number of \( r \) sized subsets of
\( C \) (number of users) we can choose from a particular \( S \). Thus rate \( R = \left( \frac{C}{r} \right) \) and the coding gain in this scheme is \( \left( \frac{t+r}{r} \right) \)

**Remark 2:** The proposed scheme can be designed for any integer \( t = \left[ \frac{CM}{N} \right] \), and the rate points in between can be achieved through memory sharing. So the proposed scheme exists for any multi-access network with \( N \) files, \( C \) caches equipped with memories of size \( M \) file units each and access degree \( r \).

**Remark 3:** Note that in this setup, since every user is associated with a unique subset of caches, even if some users leave the network, the transmissions involving the existing users continue to hold. The only constraint is that, any user in the network should be uniquely associated with an \( r \) subset of caches. So our setup is dynamic in the sense that users can join and leave the system whenever they want, provided the association with the caches is unique. \( \left( \frac{t}{r} \right) \) is simply the maximum number of users associated with distinct \( r \) subsets of caches that can be supported by the described the placement and delivery scheme.

**Remark 4:** In all the multi-access schemes in literature except the one derived from cross resolvable designs, the \( r \) caches connected to any user, store disjoint contents. As a consequence, it is seen that in schemes with cyclic setups, the rate \( R \) becomes 0 when \( \frac{M}{N} \geq \frac{1}{r} \), since the user gets all the \( N \) files from the \( r \) caches connected to it. However, in these schemes, it is possible to enforce the constraint of making \( r \) caches store disjoint contents since the number of users in the network is only \( C \). In the setup considered here, since the number of users is more than \( C \), the same subfiles can be stored in multiple caches connected to a user and hence rate does not become 0, when \( \frac{M}{N} \geq \frac{1}{r} \). Though this redundancy does appear like a waste of memory, since the number of users supported is large in the network, multi-casting opportunities increase resulting in low per user rates.

**Remark 5:** In the proposed scheme, the contents placed in the caches connected to any user in the setup are disjoint when \( t = 1 \), since the contents placed in the caches itself is disjoint in this special case.

\[ Z_i \cap Z_j = \emptyset, \text{ } \forall i,j \in C \]

It can be noted that \( t = 1 \), is the only case when contents connected to a user become disjoint, like in the existing schemes in literature where caches connected to a user always hold disjoint contents with the exception of the CRD coded caching scheme.

### A. Examples

In the examples below, we assume that the users are numbered by lexicographically ordered \( r \) subsets. For example if \( r = 2 \), \( C = 4 \), then User 1 corresponds to the subset \{1,2\}, User 2 to \{1,3\}, User 3 to \{1,4\}, User 4 to \{2,3\}, User 5 to \{2,4\}, and User 6 to \{3,4\}. The request vector \( (d_1,d_2,\ldots,d_K) \in N^K \) denotes the tuple containing demands of the users \( 1,2,\ldots,K \). The following three examples correspond to the cases \( t = 1 \), \( t \neq r \), and \( t = r \) respectively.

**Example 1:** Consider a multi-access setup with \( C = 4 \) caches, and cache access degree \( r = 2 \). We allow all possible combinations of access to \( r \) caches. So number of users \( K = \left( \frac{C}{r} \right) = \left( \frac{4}{2} \right) = 6 \). For \( t = 1 \) the subpacketization is \( F = \left( \frac{C}{r} \right) = \left( \frac{4}{2} \right) = 4 \). The subfiles are \( W_{i,1}, W_{i,2}, W_{i,3}, W_{i,4}, \forall \ i \in [N] \). The cache placement is:

\[
\begin{align*}
Z_1 &= \{W_{1,1},W_{2,1},W_{3,1},W_{4,1},W_{5,1},W_{6,1}\} \\
Z_2 &= \{W_{1,2},W_{2,2},W_{3,2},W_{4,2},W_{5,2},W_{6,2}\} \\
Z_3 &= \{W_{1,3},W_{2,3},W_{3,3},W_{4,3},W_{5,3},W_{6,3}\} \\
Z_4 &= \{W_{1,4},W_{2,4},W_{3,4},W_{4,4},W_{5,4},W_{6,4}\}
\end{align*}
\]

Let the request vector be \( (1,2,3,4,5,6) \). The transmissions are:

\[
\begin{align*}
Y_{(1,2,3)} &= W_{1,3} \oplus W_{2,2} \oplus W_{4,1} \\
Y_{(1,2,4)} &= W_{1,4} \oplus W_{5,1} \oplus W_{3,2} \\
Y_{(1,3,4)} &= W_{2,4} \oplus W_{3,3} \oplus W_{6,1} \\
Y_{(2,3,4)} &= W_{5,3} \oplus W_{6,2} \oplus W_{4,4}
\end{align*}
\]

**Example 2:** Consider a multi-access setup with \( C = 5 \) caches, and cache access degree \( r = 3 \), \( t = 2 \). Number of users \( K = \left( \frac{C}{r} \right) = \left( \frac{5}{3} \right) = 10 \) and subpacketization is \( F = \left( \frac{C}{r} \right) = \left( \frac{5}{3} \right) = 10 \). The subfiles are \( W_{i,(1,2)}, W_{i,(1,3)}, W_{i,(1,4)}, W_{i,(1,5)}, W_{i,(2,3)}, W_{i,(2,4)}, W_{i,(2,5)}, W_{i,(3,4)}, W_{i,(3,5)}, W_{i,(4,5)}, \forall \ i \in [N] \). The cache placement is:

\[
\begin{align*}
Z_1 &= \{W_{i,(1,2)},W_{i,(1,3)},W_{i,(1,4)},W_{i,(1,5)}, \forall \ i \in [N]\} \\
Z_2 &= \{W_{i,(1,2)},W_{i,(1,3)},W_{i,(2,4)},W_{i,(2,5)}, \forall \ i \in [N]\} \\
Z_3 &= \{W_{i,(1,3)},W_{i,(2,3)},W_{i,(3,4)},W_{i,(3,5)}, \forall \ i \in [N]\} \\
Z_4 &= \{W_{i,(1,4)},W_{i,(2,4)},W_{i,(3,4)},W_{i,(4,5)}, \forall \ i \in [N]\} \\
Z_5 &= \{W_{i,(1,5)},W_{i,(2,5)},W_{i,(3,5)},W_{i,(4,5)}, \forall \ i \in [N]\}
\end{align*}
\]

Let the request vector be \( (1,2,3,4,5,6,7,8,9,10) \). The transmissions are:

\[
\begin{align*}
Y_{(1,2,3,4,5)} &= W_{1,(4,5)} \oplus W_{2,(3,5)} \oplus W_{3,(3,4)} \oplus W_{4,(2,5)} \oplus W_{5,(2,4)} \oplus W_{6,(2,3)} \oplus W_{7,(1,5)} \oplus W_{8,(1,4)} \oplus W_{9,(1,3)} \oplus W_{10,(1,2)}
\end{align*}
\]
Example 3: Consider a multi-access setup with $C = 5$ caches, and cache access degree $r = 2$, $t = 2$. Number of users $K = \binom{5}{2} = \binom{3}{1} = 10$ and subpacketization is $F = \binom{C}{t} = \binom{5}{2} = 10$. The subfiles are $W_i(1,2), W_i(1,3), W_i(1,4), W_i(1,5), W_i(2,3), W_i(2,4), W_i(2,5), W_i(3,4), W_i(3,5), W_i(4,5), \forall i \in [N]$. The cache placement is:

$Z_1 = \{W_i(1,2), W_i(1,3), W_i(1,4), W_i(1,5), \forall i \in [N]\}$

$Z_2 = \{W_i(1,2), W_i(1,3), W_i(2,3), W_i(2,5), \forall i \in [N]\}$

$Z_3 = \{W_i(1,3), W_i(1,4), W_i(2,3), W_i(3,5), \forall i \in [N]\}$

$Z_4 = \{W_i(1,4), W_i(1,2), W_i(2,4), W_i(3,4), W_i(4,5), \forall i \in [N]\}$

$Z_5 = \{W_i(1,5), W_i(1,2), W_i(3,4), W_i(4,5), \forall i \in [N]\}$

Let the request vector be $(1, 2, 3, 4, 5, 6, 7, 8, 9, 10)$. The transmissions are:

\[
Y_{1,1,3,5} = W_{1}(3,4) \oplus W_{2}(2,4) \oplus W_{3}(2,3) \oplus W_{5}(1,4) \oplus W_{6}(1,3) \oplus W_{8}(1,2)
\]

\[
Y_{1,2,3,5} = W_{1}(3,5) \oplus W_{2}(2,5) \oplus W_{4}(2,3) \oplus W_{5}(1,5) \oplus W_{7}(1,3) \oplus W_{9}(1,2)
\]

\[
Y_{1,2,4,5} = W_{1}(4,5) \oplus W_{3}(2,5) \oplus W_{4}(2,4) \oplus W_{6}(1,5) \oplus W_{7}(1,4) \oplus W_{10}(1,2)
\]

\[
Y_{1,3,4,5} = W_{2}(4,5) \oplus W_{3}(3,5) \oplus W_{4}(3,4) \oplus W_{8}(1,5) \oplus W_{9}(1,4) \oplus W_{10}(1,3)
\]
\[ Y_{(2,3,4,5)} = W_{6,(4,5)} \oplus W_{6,(3,5)} \oplus W_{4,(3,4)} \oplus W_{8,(1,5)} \oplus W_{9,(2,4)} \oplus W_{10,(2,3)} \]

III. PERFORMANCE ANALYSIS

In this section we compare the performance of the proposed scheme with the schemes available in the literature for multi-access coded caching.

A. Comparison with the HKD Scheme [12]

In Fig. 2a the proposed scheme and the HKD scheme are compared in terms of per user rate \( R_K \) with respect to \( M_N \), for \( M_N < \frac{1}{r} \), and for different values of \( r \). The expression for \( R \) of centralized equivalent of the HKD scheme is given in (1).

Note that \( r \) should divide \( C \) for the centralized HKD scheme to exist. The number of users \( K = C \) in the HKD scheme.

Fig. 2b depicts variation of per user rate \( R_K \) with respect to \( M_N \) for the case when \( M_N \geq \frac{1}{r} \). From Fig. 2a it is seen that the proposed scheme achieves lower per user rate than the HKD scheme when \( M_N < \frac{1}{r} \). For the case, \( M_N \geq \frac{1}{r} \), in Fig. 2b it is is seen that the rate of the HKD scheme becomes 0, while the per user rate of proposed scheme is slightly higher. However this is expected due to the cyclic wraparound setup considered in the HKD scheme and the placement in the caches such that the user gets all the \( N \) files when \( \frac{M_N}{N} \geq \frac{1}{r} \).

For the centralized equivalent of the HKD scheme, \( F \) is given by \( r \left( \frac{C}{r} t \right) \).

In Fig. 3a the variation of subpacketization \( F \), with respect to \( K \) is studied, keeping access degree \( r \) constant for both schemes and the comparison is made for different values of \( t \). In Fig. 3b the variation of subpacketization \( F \), with respect to \( K \) is studied, keeping \( t = \frac{C M_N}{N} \) constant for both schemes and the comparison is made for different values of \( r \). It is seen that the subpacketization levels of proposed scheme is significantly lower than that of the HKD scheme for large number of users.

Figure 4: Comparison with the SPE scheme for number of caches, \( C = 18 \) for different values of \( r \) i.e \( r \in \{2, 3, 4, 5\} \).

Figure 5: Comparison with the SPE scheme for different values of \( r \) i.e \( r \in \{2, 3, 4, 5\} \) for the fraction of each file stored at each cache the same in both the schemes i.e. \( \frac{M_N}{N} = \frac{2}{7} \).

Figure 6: Comparison with the RK scheme for the number of caches, \( C = 12 \) for different values of \( r \) i.e \( r \in \{6, 7, 8, 9, 10, 11\} \).
B. Comparison with the SPE Scheme [16]

In Fig. 4, the proposed scheme and the SPE scheme for $K_M N = 2$ are studied in terms of per user rate $R_K$ with respect to $M_N$, by keeping access degree, $r = 2$ for both the schemes. The expression for rate of the SPE scheme for $K_M N = 2$ is given in Theorem 1 of [16]. For per user rate, it is normalized by number of users (For the SPE scheme $K = C$). Since this scheme exists only for $K_M N = 2$, only these specific points are plotted for comparison. From Fig. 4 it is seen that proposed scheme supports lower per user rates.

The next comparison is in terms of subpacketization. In Fig. 5, variation of $F$ for the two schemes with respect to number of users $K$ is studied, for different values of $r$. It is seen that apart from supporting large number of users, lower subpacketization levels are obtained from the proposed scheme when compared to the SPE scheme.

In the plot shown above, the two schemes are compared for different values of $r i.e r \in \{2, 3, 4, 5\}$, $t = 2$. Figure 7: Comparison with the RK scheme in terms of subpacketization level ($F$).

For the case $r = C - 1$, the authors in [16] propose an alternate scheme that achieves optimality, in the sense that the rate here exactly matches that of MAN scheme with each user having access to memory $rM$ each. Through search, we see that certain parameters corresponding to this case, can result in our scheme performing bad in terms of per user rate and subpacketization when compared to SPE scheme. Some of these parameters are depicted in Table I. In [16], for the scheme corresponding to $r = C - 1\, t = C$, our subpacketization is $F_{SP} = (\frac{C}{t})$, while our subpacketization is $F_{SP} = (\frac{C}{t})$. When $r > t$, then proposed scheme is better in terms of subpacketization with respect to number of users. When $r = t$, both have same performance and when $r < t$, then SPE scheme is better in terms of subpacketization with respect to number of users. For the case when $r = C - 1, t = 1$, our setup matches cyclic setup and $r = C - 1$. In this case the per user rate and subpacketization are same for both the schemes. For the case when $r = 1, t = C - 1$, our setup matches the Maddah Ali Niesen scheme [11] for dedicated caches and in this case also, the per user rate and subpacketization are exactly the same for both schemes. It is interesting to observe that there does exist some parameters with $C - rt = 1$, where proposed scheme performs better than SPE scheme in terms of per user rate.

Some of these parameters are given in Table II.

| C  | r  | t  | $R_{prop}$ | $R_{SPE}$ |
|----|----|----|------------|------------|
| 5  | 2  | 2  | 1.25       | 1.7        |
| 7  | 3  | 2  | 1.4        | 1.24       |
| 7  | 2  | 3  | 1.4        | 1.7        |
| 9  | 4  | 2  | 1.5        | 1.727      |
| 9  | 2  | 4  | 1.5        | 1.268      |
| 10 | 3  | 3  | 1.458      | 1.268      |
| 11 | 5  | 2  | 1.571      | 1.727      |
| 11 | 2  | 5  | 1.571      | 1.75       |
| 13 | 6  | 2  | 1.625      | 1.064      |
| 13 | 4  | 3  | 1.418      | 1.064      |
| 13 | 3  | 4  | 1.418      | 1.75       |
| 13 | 2  | 6  | 1.625      | 1.192      |
| 15 | 7  | 2  | 1.667      | 1.192      |
| 15 | 2  | 7  | 1.667      | 1.769      |
| 16 | 5  | 3  | 1.347      | 1.769      |
| 16 | 3  | 5  | 1.347      | 1.786      |
Table II: Certain parameters for which proposed scheme is better compared to SPE scheme

| C  | r  | t  | \( \frac{R_{lb}}{R_{Prop}} \) |
|----|----|----|---------------------|
| 25 | 6  | 4  | 1.097               |
| 25 | 4  | 6  | 1.097               |
| 26 | 5  | 5  | 1.205               |
| 29 | 7  | 4  | 1.274               |
| 29 | 4  | 7  | 1.274               |
| 31 | 10 | 3  | 1.006               |
| 31 | 6  | 5  | 1.537               |
| 31 | 3  | 10 | 1.006               |
| 31 | 8  | 4  | 1.47                |
| 33 | 4  | 8  | 1.47                |
| 34 | 11 | 3  | 1.064               |
| 36 | 7  | 5  | 1.94                |
| 36 | 5  | 7  | 1.94                |
| 37 | 12 | 3  | 1.123               |
| 37 | 9  | 4  | 1.685               |
| 37 | 6  | 6  | 2.131               |
| 37 | 4  | 9  | 1.685               |
| 37 | 3  | 12 | 1.123               |
| 40 | 13 | 3  | 1.182               |

C. Comparison with the RK Scheme [15]

In this subsection, the proposed scheme is compared with the RK scheme by varying \( \frac{M}{N} \) and noticing its effect on per user rate. For the RK scheme the normalized lower bound of rate \( R_{lb} \) is plotted. The expression for \( R_{lb} \) is taken from Theorem 3 in [15]. Since this lower bound is valid for \( r \geq \frac{C}{2} \), the comparison in Fig. 6 holds only for \( r \geq \frac{C}{2} \). Also \( K = C \) in the RK scheme. The schemes are compared for different values of \( r \). The per user rate for the proposed scheme is found to be better than that of the RK scheme for the range \( \frac{M}{N} < \frac{1}{2} \) from Fig. 6a.

In Fig. 7a the two schemes are compared keeping \( \frac{M}{N} \) constant i.e. \( \frac{M}{N} = \frac{1}{2} \). The two schemes have been compared for different values of \( r \). In Fig. 7b the two schemes are compared by keeping the access degree same in both the schemes i.e. \( r = 2 \). The two schemes have been compared for different values of \( t \). From these plots it can be concluded that significantly low subpacketization levels can be attained with the proposed scheme for large number of users compared with the RK scheme.

D. Comparison with the CRD Scheme [17]

The scheme from [17] derived from the cross resolvable designs from affine planes is compared with proposed scheme in this section. The rate per user for the CRD scheme from affine planes is \( R_{CRD} = \frac{(n-1)^2}{4n^2} \) where \( n \) is a prime or prime power. It can be seen from Fig. 8 that the rate per user supported, in the proposed scheme is significantly less compared to the scheme derived from CRD.

From [17], \( \frac{M}{N} = \frac{1}{n} \), \( C = n(n+1) \), \( F = n^2 \) and \( K = n^3(n+1)^2 \) for the multi-access scheme from CRDs derived from affine planes, where \( n \) is a prime or prime power. In Fig. 9a, the number of caches \( C \) and memory fraction \( \frac{M}{N} \) is kept same for both schemes, and the two schemes are compared in terms of subpacketization levels \( F \). Since \( C \) and \( \frac{M}{N} \) are dependent only on the parameter \( n \), for the CRD scheme from affine planes, the subpacketization values are plotted with respect to \( n \). In Fig. 9b the subpacketization levels are plotted with respect to number of users \( K \). From Fig. 9a and Fig. 9b it can be concluded that the proposed scheme does not perform well in terms of subpacketization levels when compared with the CRD scheme. But this is explained by the fact that the CRD scheme loses out in rate and gains in terms of subpacketization. Also, the number of users supported in the proposed scheme is more than that of the CRD scheme.

E. Comparison with the Cheng-Liang-Wan-Zhang-Caire (CLWZC) Scheme [18]

In this subsection, the proposed scheme is compared with the CLWZC [18] scheme by varying \( \frac{M}{N} \) and noticing...
its effect on per user rate. The expression of rate $R$ for [18] scheme is taken from Theorem 1 in [18]. The schemes are compared for different values of $r$. The per user rate for the proposed scheme is found to be better than that of the CLWZC scheme for the range $\frac{M}{N} < \frac{1}{r}$ from Fig. 10a.

In Fig. 11b the two schemes are compared keeping $\frac{M}{N}$ constant i.e. $\frac{M}{N} = \frac{2}{C}$. The two schemes have been compared for different values of $r$. In Fig. 11a, the two schemes are compared by keeping the access degree same in both the schemes i.e. $r = 2$. The two schemes have been compared for different values of $t$. From these plots it can be seen that significantly low subpacketization levels can be attained with the proposed scheme when compared with the CLWZC scheme.

In Table III the proposed scheme is compared with the CLWZC scheme for $t = \frac{CM}{N} = 1$. In can be observed that for $t = 1$ and the access degree $r = C - 2$ the proposed scheme achieves lower subpacketization level ($F$) and supports larger number of users ($K$) than the CLWZC scheme for the same fraction of each file at each cache $\frac{M}{N}$, fraction of each file each user has access to $\frac{M}{N}$ and rate ($R$).

$F$. Comparison with the SR Scheme 1 [19]

Here, we compare our scheme to the SR Scheme 1. The per user rate for the proposed scheme is found to be better than that of the SR scheme 1 scheme for the range $\frac{M}{N} < \frac{1}{r}$ from Fig. 12.

The subpacketization of the SR scheme 1 is atmost $F_{SPE} = C^2 = K^2$ as $C = K$, while ours is $\binom{C}{t}$ which can be less than, equal to, or greater than the number of users in the setup $\binom{C}{r}$. So atleast for some choices of $t$ and $r$ the proposed scheme is better than the SR scheme 1 in terms of subpacketization.
Table III: Comparison between CLWZC and Proposed scheme for $t = \frac{CM}{N} = 1$.

| Parameters                        | CLWZC Scheme | Proposed Scheme |
|-----------------------------------|--------------|-----------------|
| Number of Caches ($C$)            | $C$          | $C$             |
| Number of Caches a user has access to ($r$) | $C - 2$      | $C - 2$         |
| Number of Users ($K$)             | $C$          | $(\binom{C}{2})/2$ |
| Fraction of each file at each cache ($\frac{M}{N}$) | $\frac{1}{C}$ | $\frac{1}{C}$  |
| Fraction of each file each user has access to ($\frac{C-2}{C}$) | $\frac{C-2}{C}$ | $\frac{C-2}{C}$ |
| Subpacketization level ($C(\frac{C-1}{t})$) | $3C$         | $C$             |
| Rate ($R$)                        | $\frac{C-t}{t+1} = 1$ | $\frac{C}{(t)} = 1$ |

When $r = t$ for instance, the proposed scheme is better in terms of subpacketization ($F$).

Note that in this subsection, we have not considered the case $r = \frac{CM}{N}$, since it is already covered under comparison with SPE scheme.

G. Comparison with the SR Scheme 2 [20]

Here, we compare our scheme to the SR Scheme 2. The per user rate for the proposed scheme is found to be better than that of the SR scheme 2 scheme for the range $\frac{M}{N} < \frac{1}{r}$ from Fig. 13. The subpacketization of the SR scheme 2 is $F_{SPE} = C = K$, while ours is $\binom{C}{2}$ which can be greater than, less than or equal to $\binom{C}{t}$, which says that our scheme can be better in terms of subpacketization than the SR scheme 2 for some choices of $r$ and $t$. When $r = t$, both schemes have the same subpacketization level. Figure 12: Comparison with the SR scheme 1 for the number of caches, $C = 18$ for different values of $r$ i.e $r \in \{2, 4, 6, 8, 10, 12, 14, 16\}$. 

(a) $\frac{M}{N} < \frac{1}{r}$

(b) $\frac{M}{N} \geq \frac{1}{r}$
same performance in terms of subpacketization with respect to number of users.

IV. CONCLUSION

We conclude that the proposed scheme is better than most existing schemes in literature since the number of users ($K$) supported is very large at relatively low subpacketization ($F$) levels. When $\frac{M}{N} < \frac{1}{r}$ and $r \neq \frac{C-1}{C M}$, the rate per user achieved is lesser than that of existing schemes. When $\frac{M}{N} < \frac{1}{r}$ and $r = \frac{C-1}{C M}$, the performance with respect to per user rate may be better or worse compared to existing schemes depending on the values of $C$, $r$ and $C \frac{M}{N}$. The proposed scheme exists for any multi-access network with $N$ files, $C$ caches equipped with memories of size $M$ file units each and access degree $r$. The scheme can be designed for any integer $\frac{C M}{N}$, and the rate points in between can be achieved through memory sharing, allowing convenience in designing.

ACKNOWLEDGMENT

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.

REFERENCES

[1] M. A. Maddah-Ali and U. Niesen, “Fundamental limits of caching,” IEEE Trans. Inf. Theory, vol. 60, no. 5, pp. 2856–2867, May 2014.
[2] Q. Yan, M. Cheng, X. Tang and Q. Chen, “On the Placement Delivery Array Design for Centralized Coded Caching Scheme,” in IEEE Transactions on Information Theory, vol. 63, no. 9, pp. 5821-5833, Sept. 2017.
[3] P. Krishnan, “Coded Caching via Line Graphs of Bipartite Graphs,” 2018 IEEE Information Theory Workshop (ITW), Guanzhou, 2018, pp. 1-5.
[4] L. Tang and A. Ramamoorthy, “Coded Caching Schemes With Reduced Subpacketization From Linear Block Codes,” in IEEE Transactions on Information Theory, vol. 64, no. 4, pp. 3099-3120, April 2018.
[5] S. Agrawal, K. V. Sushena Sree and P. Krishnan, “Coded Caching based on Combinatorial Designs,” 2019 IEEE International Symposium on Information Theory (ISIT), Paris, France, 2019, pp. 1227-1231.
[6] J. Wang, M. Cheng, Q. Yan and X. Tang, “Placement Delivery Array Design for Coded Caching Scheme in D2D Networks,” in IEEE Transactions on Communications, vol. 67, no. 5, pp. 3388-3395, May 2019.
[7] Q. Yan, M. Wigger and S. Yang, “Placement Delivery Array Design for Combination Networks with Edge Caching,” 2018 IEEE International Symposium on Information Theory (ISIT), Vail, CO, 2018, pp. 1555-1559.
[8] B. Asadi and L. Ong, “Centralized Caching with Shared Caches in Heterogeneous Cellular Networks,” 2019 IEEE 20th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC), Cannes, France, 2019, pp. 1-5.
[9] E. Parrinello, A. Unsal, and P. Elia, “Coded caching with shared caches: Fundamental limits with uncoded prefetching,” arXiv preprint arXiv:1809.09422 2018.
[10] A. M. Ibrahim, A. A. Zewail and A. Yener, “Coded Placement for Systems with Shared Caches,” ICC 2019 - 2019 IEEE International Conference on Communications (ICC), Shanghai, China, 2019, pp. 1-6.
[11] U. Niesen and M. A. Maddah-Ali, “Coded Caching With Nonuniform Demands,” in IEEE Transactions on Information Theory, vol. 63, no. 2, pp. 1146-1158, Feb. 2017.
[12] 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. 3108-141, 2017.
[13] K. S. Reddy and N. Karamchandani, “On the Exact Rate-Memory Trade-off for Multi-access Coded Caching With Uncoded Placement,” 2018 International Conference on Signal Processing and Communications (SPCOM), Bangalore, India, 2018, pp. 1-5.
[14] K. S. Reddy and N. Karamchandani, “Rate-memory trade-off for multi-access coded caching with uncoded placement,” in IEEE International Symposium on Information Theory (ISIT), 2019.
[15] 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.
[16] Berksan Serbetci, Emanuele Parrinello and Petros Elia, “Multi-access coded caching : gains beyond cache-redundancy,” IEEE Information Theory Workshop, Visby, Gotland, 2019.
[17] Digvijay Katyal, Pooja Nayak Mand B Sundar Rajan, “Multi-access Coded Caching Schemes From Cross Resolvable Designs,” Available on arXiv preprint arXiv:2005.13731 [cs.IT]. Accepted for publication in IEEE Transactions on Communications.
[18] Minquan Cheng, Dequan Liang, Kai Wan, Mingming Zhang, and Giuseppe Caire, “A Novel Transformation Approach of Shared-link Coded Caching Schemes for Multiaccess Networks,” Available on arXiv:2012.04483v2 [cs.IT].
[19] S. Sasi and B. S. Rajan “An Improved Multi-access Coded Caching With Uncoded Placement,” Available on arXiv:2009.05377 [cs.IT].
[20] S. Sasi and B. S. Rajan “Multi-access Coded Caching Scheme with Linear Sub-packetization using PDAs,” Available on arXiv:2102.06616 [cs.IT].