Improved Multi-access Coded Caching Schemes From Cross Resolvable Designs

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—**Recently multi-access coded caching schemes with number of users different from the number of caches obtained from a special class of resolvable designs called Cross Resolvable Designs (CRDs) have been reported and a new performance metric called rate-per-user has been introduced by Digvijay et al ("Multi-Access Coded Caching Schemes From Cross Resolvable Designs" in *IEEE Transactions on Communications*, May 2021). In this paper, we present a generalization of this work resulting in multi-access coded caching schemes with improved rate-per-user.**

I. INTRODUCTION

Coded Caching has drawn considerable attention after the work of [1], which proposed a cache placement scheme for uncoded content storage and a coded multicasting delivery strategy that helped to provide both global and local caching gain for a significant delivery rate reduction. The main focus of coded caching is in designing schemes with reduced rate and practical subpacketization levels. Over the years there have been many approaches aimed at reducing subpacketization levels that involved developing coded caching schemes from resolvable designs from linear block codes [2], block designs [3], placement delivery arrays [4]. While the focus has been mostly on setups with users equipped with dedicated caches, the setups where users have to share caches is also of utmost interest [5] [6] [7]. Another interesting setup is where a user has access to multiple caches and vice versa. This setup is motivated from the fact that placing cache at local access points could significantly reduce the base station transmission rate, with each user being able to access content at multiple access points along with the base station broadcast.

The setting of multi-access setup was first considered in the work of [8], in which the setting where K users and K caches, where each user is associated to z nearby caches in a cyclic way was dealt with. Later on, many schemes have been proposed for this multi-access setup [9] [10] [11]. The work of [12] gives a multi-access setup through a combinatorial design called Cross Resolvable Designs (CRD), which was found to support a large number of users at low subpacketization levels. A metric called per-user-rate or rate per user was also introduced in [12], which allowed to compare different coded caching setups. The scheme proposed in [12] was found to provide lower per-user-rates than Maddah Ali Niesen (MaN) scheme for the high memory regime.
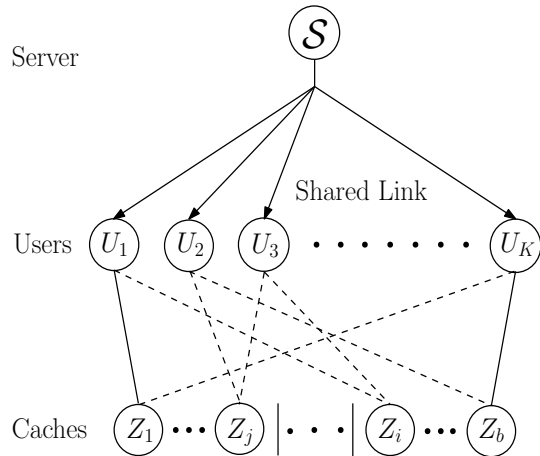

Figure 1. Problem setup for multi-access coded caching with K users, b caches and each user, connected to z caches.

A. Multi-access Coded Caching - System Model

Fig. 1 shows a multi-access coded caching system with a unique server $\mathcal{S}$ storing N files $W_1, W_2, W_3, \ldots, W_N$ each of unit size. There are K users in the network connected via an error free shared link to the server $\mathcal{S}$. The set of users is denoted by $\mathcal{K}$. There are b number of helper caches each of size M files. Each user has access to z out of the b helper caches. Let $\mathcal{Z}_k$ denote the content in the k-th cache. It is assumed that each user has an unlimited capacity link to the caches it is connected to. The setup where users are equipped with dedicated caches can be viewed as a special case of the scheme corresponding to Fig. 1 with $b = K$ and $z = 1$.

In any coded caching scheme there are two phases: the placement phase and the delivery phase. During the placement phase certain parts of each file are stored in each cache which is carried out during the off-peak hours. During the peak hours each user demands a file and the server broadcasts coded transmissions such that each user can recover its demand by combining the received transmissions with what has been stored in the caches it has access to. This is called delivery phase. The coded caching problem is to jointly design the placement and the delivery with minimal number of transmissions to satisfy the demands of all the users. The amount of transmissions used in the unit of files is called the *rate* or the *delivery time*. Subpacketization level is the number of packets that a file is divided into. Coding gain is defined as the number

of users benefited in a single transmission.

B. Contributions

This work is a generalization of the multi-access scheme in [12]. A modification in the set-up of [12] is introduced which enables having a larger number of users than in [12]. This allows to achieve lower per-user-rates than [12] at the same subpacketization levels.

C. Preliminaries

In this section, we review some of the definitions in [13] and [12]. We use a class of combinatorial designs called resolvable designs [13] to specify placement in the caches.

Definition 1: [2] A design is a pair $(X, \mathcal{A})$ such that

- X is a finite set of elements called points, and
- $\mathcal{A}$ is a collection of nonempty subsets of X called blocks, where each block contains the same number of points.

Definition 2: [2] A parallel class $\mathcal{P}$ in a design $(X, \mathcal{A})$ is a subset of disjoint blocks from $\mathcal{A}$ whose union is X. A partition of $\mathcal{A}$ into several parallel classes is called a resolution, and $(X, \mathcal{A})$ is said to be a resolvable design if A has at least one resolution.

Example 1: Consider a block design specified as follows.

$$X = \{1, 2, 3, 4\}, \text{ and}$$
$$\mathcal{A} = \{\{1, 2\}, \{1, 3\}, \{1, 4\}, \{2, 3\}, \{2, 4\}, \{3, 4\}\}.$$

It can be observed that this design is resolvable with the parallel classes $\mathcal{P}_1 = \{\{1, 2\}, \{3, 4\}\}$, $\mathcal{P}_2 = \{\{1, 3\}, \{2, 4\}\}$, and $\mathcal{P}_3 = \{\{1, 4\}, \{2, 3\}\}$. Note that $\mathcal{P}_1$, $\mathcal{P}_2$, $\mathcal{P}_3$ forms a partition of $\mathcal{A}$. If $\mathcal{A} = \{\{1, 2\}, \{1, 3\}, \{3, 4\}, \{2, 4\}\}$, we get another resolvable design with two parallel classes $\mathcal{P}_1$ and $\mathcal{P}_2$.

Example 2: Consider a block design specified as follows.

$$X = \{1, 2, 3, 4, 5, 6\}, \text{ and}$$
$$\mathcal{A} = \{\{1, 2, 3\}, \{4, 5, 6\}, \{1, 4, 5\}, \{2, 3, 6\}\}.$$

It can be observed that this design is resolvable with two parallel classes: $\mathcal{P}_1 = \{\{1, 2, 3\}, \{4, 5, 6\}\}$ and $\mathcal{P}_1 = \{\{1, 4, 5\}, \{2, 3, 6\}\}$.

For a given resolvable design $(X, \mathcal{A})$ if $|X| = v$, $|\mathcal{A}| = b$, block size is k and number of parallel classes is r, then there are exactly $\frac{b}{r}$ blocks in each parallel class. Since the blocks in each parallel class are disjoint, therefore number of blocks in each parallel class $= \frac{b}{r} = \frac{v}{k}$.

D. Cross Resolvable Design (CRD)

*Definition 3 (**Cross Intersection Number**):* For any resolvable design $(X, \mathcal{A})$ with r parallel classes, the ith cross intersection number, μ_i where $i \in \{2, 3, \ldots, r\}$, is defined as the cardinality of intersection of i blocks drawn from any i distinct parallel classes, provided that, this value remains same $(\mu_i \neq 0)$, for all possible choices of blocks. For instance, in Example 1, $\mu_2 = 1$ and μ_3 does not exist.

*Definition 4 (**Cross Resolvable Design**):* For any resolvable design $(X, \mathcal{A})$, if there exist at least one $i \in \{2, 3, \ldots, r\}$ such that the ith cross intersection number μ_i exists, then

the resolvable design is said to be a Cross Resolvable Design (CRD).

Note that the resolvable design in *Example 2* is not a CRD as μ_2 does not exist.

Example 3: For the resolvable design $(X, \mathcal{A})$ with

$$X = \{1, 2, 3, 4, 5, 6, 7, 8, 9\}, \text{ and}$$
$$\mathcal{A} = \{\{1, 2, 3\}, \{4, 5, 6\}, \{7, 8, 9\},$$
$$\{1, 4, 7\}, \{2, 5, 8\}, \{3, 6, 9\}\},$$

the parallel classes are $\mathcal{P}_1 = \{\{1, 2, 3\}, \{4, 5, 6\}, \{7, 8, 9\}\}$, and $\mathcal{P}_2 = \{\{1, 4, 7\}, \{2, 5, 8\}, \{3, 6, 9\}\}$. It is easy to verify that $\mu_2 = 1$.

Example 4: For the resolvable design $(X, \mathcal{A})$ with

$$X = \{1, 2, 3, 4, 5, 6, 7, 8\}, \text{ and}$$
$$\mathcal{A} = \{\{1, 2, 3, 4\}, \{5, 6, 7, 8\}, \{1, 2, 5, 6\},$$
$$\{3, 4, 7, 8\}, \{1, 3, 5, 7\}, \{2, 4, 6, 8\}\},$$

the parallel classes are $\mathcal{P}_1 = \{\{1, 2, 3, 4\}, \{5, 6, 7, 8\}\}$, $\mathcal{P}_2 = \{\{1, 2, 5, 6\}, \{3, 4, 7, 8\}\}$, and $\mathcal{P}_3 = \{\{1, 3, 5, 7\}, \{2, 4, 6, 8\}\}$. We have $\mu_2 = 2$ and $\mu_3 = 1$.

II. CODED CACHING SCHEMES FROM CRDs

Given a CRD $(X, \mathcal{A})$ with v points, r parallel classes, b blocks of size k each, $b_r \stackrel{def}{=} \frac{b}{r}$ blocks in each parallel class, we choose some $z \in \{2, 3, \ldots, r\}$ such that μ_z exists. Let $\mathcal{A}_j$ denote the jth block in $\mathcal{A}$, assuming some ordering on the blocks of $\mathcal{A}$. We associate a coded caching problem with $K = \binom{r}{z}\binom{b_r}{t}^z$ number of users where, $t \in \{1, 2, \ldots, b_r\}$, N files in server database, b number of caches, $\frac{M}{N} = \frac{k}{v}$ fraction of each file at each cache and subpacketization level v. A user is connected to distinct tz caches such that these tz caches correspond to distinct t blocks from each of the distinct z parallel classes. We denote the set of K users $\mathcal{K}$ as, $\mathcal{K} := \{U_H : |H| = tz\}$ where, H is a tz sized set containing cache indices from distinct parallel classes.

A. Placement Phase

In the caching placement phase, we split each file W_i, $\forall i \in [N]$ into v non-overlapping subfiles of equal size i.e.

$$W_i = (W_{i,k} : \forall k \in [v]), \ i = 1, 2, \ldots, N.$$

The placement is as follows. In the jth cache, the indices of the subfiles stored in $\mathcal{Z}_j$ is the jth block in the design. We assume symmetric batch prefetching i.e.,

$$\mathcal{Z}_j = \{W_{ik} : k \in \mathcal{A}_j\}, \ \forall i \in \{1, 2 \ldots, N\}, \forall j \in \{1, 2 \ldots, b\}.$$

Therefore the total number of subfiles for each file in any cache is block size k of the resolvable design i.e. $\frac{M}{N} = \frac{k}{v}$.

Let M' denote the size of the memory in units of files that a user has access to. We have

$$\frac{M'}{N} = \sum_{i=1}^{tz} \frac{|\mathcal{A}_i|}{v} - \sum_{1 \le i_1 < i_2 \le tz} \frac{|\mathcal{A}_{i_1} \cap \mathcal{A}_{i_2}|}{v} + \cdots + (-1)^{s+1}$$

$$\sum_{1 \le i_1 < \cdots < i_s \le tz} \frac{|\mathcal{A}_{i_1} \cap \cdots \cap \mathcal{A}_{i_s}|}{v} + \cdots +$$

$$(-1)^{tz+1} \frac{|\mathcal{A}_1 \cap \cdots \cap \mathcal{A}_{tz}|}{v}$$

where $\mathcal{A}_i$, $i \in [tz]$ are distinct t blocks from each of distinct z parallel classes. We get,

$$\frac{M'}{N} = zt\left(\frac{M}{N}\right) - (t^2)\binom{z}{2}\left(\frac{\mu_2}{v}\right) + \cdots +$$
$$(-1)^{s+1}(t^s)\binom{z}{s}\left(\frac{\mu_s}{v}\right) + \cdots + (-1)^{z+1}(t^z)\left(\frac{\mu_z}{v}\right),$$

which simplifies to

$$\frac{M'}{N} = \frac{ztM}{N} + \sum_{s=2}^{z}(-1)^{s+1}(t^s)\binom{z}{s}\left(\frac{\mu_s}{v}\right).$$

B. Delivery Phase

For delivery, the users are arranged in lexicographical order of their indices S, establishing a one-to-one correspondence with the set $\{1, 2, \ldots, K\}$. We focus our attention to the case where user demands are distinct. At the beginning of the delivery phase, each user requests one of the N files. Let the demand vector be denoted by $\mathbf{d} = (d_1, d_2, \ldots, d_K)$.

The delivery steps are presented as an algorithm in **Algorithm 1**, the proof of correctness of which is very long and hence is given in [14].

Theorem 1: For N files and K users each with access to zt caches of size M in the considered caching system, if $N \ge K$ and for the distinct demands by the users, the proposed scheme achieves the rate R given by

$$R = \frac{\mu_z \binom{b_r}{t+1}^z \binom{r}{z}}{v}$$

Proof: The first **for** loop (Line 1) of the delivery algorithm runs $\binom{r}{z}$ times. The second **for** loop (Line 5) of the delivery algorithm runs $\binom{b_r}{t+1}^z$ times. The transmit step of the delivery algorithm runs μ_z times. So we see that totally there are $\frac{\mu_z \binom{b_r}{t+1}^z \binom{r}{z}}{v}$ transmissions and the subpacketization level is v. ∎

Lemma 1: The number of users benefited in each transmission, known in the literature as coding gain and denoted by g, is given by $g = (t+1)^z$.

Proof: From second **for** (Line 5) of the delivery algorithm it can be observed that, there are totally $(t+1)^z$ users benefited from a transmission. So the coding gain by definition is $(t+1)^z$. ∎

Remark 1: In the proposed scheme, when $t = 1$, the scheme in [12] is obtained.

Algorithm 1 Delivery Algorithm

1: **for** $u = 1$ to $u = \binom{r}{z}$ **do**
2:     Choose any z parallel classes out of r parallel classes
3:     which is different from sets chosen before.
4:     Let this set be

$$\mathcal{P}_1 = \{C_{1,1}, C_{1,2}, \ldots\ldots, C_{1,b_r}\},$$
$$\mathcal{P}_2 = \{C_{2,1}, C_{2,2}, \ldots\ldots, C_{2,b_r}\},$$
$$\vdots$$
$$\mathcal{P}_z = \{C_{z,1}, C_{z,2}, \ldots\ldots, C_{z,b_r}\}.$$

5:     **for** $v = 1$ to $v = \binom{b_r}{t+1}^z$ **do**
6:         Choose $t+1$ of blocks from each of the parallel
7:         classes $\mathcal{P}_1, \mathcal{P}_2, \ldots, \mathcal{P}_z$.
8:         This set of $(t+1)z$ blocks must be different from
9:         the ones chosen before.
10:        Let the chosen set be $C_{1,i_{1_0}}, C_{1,i_{1_1}}, \ldots, C_{1,i_{1_t}}$,
11:        $C_{2,i_{2_0}}, C_{2,i_{2_1}}, \ldots, C_{2,i_{2_t}} \ldots, C_{z,i_{z_0}}, C_{z,i_{z_1}},$
12:        $\ldots, C_{z,i_{z_t}},$
13:        where $i_{s_k} \in [1, b_r]$ and $i_{s_k} \ne i_{s_{k'}}$, $\forall k, k' \in [0, t]$
14:        and $\forall s \in [1, z]$.
15:        There are $(t+1)^z$ users corresponding to the
16:        $(t+1)z$ blocks chosen above.
17:        Denoting this set of user indices by $\mathcal{X}$.
18:        **Calculate:** Calling the user connected to the set of
19:        caches
20:        $C_{1,a_{1_1}}, C_{1,a_{1_2}}, \ldots, C_{1,a_{1_t}}, C_{2,a_{2_1}}, C_{2,a_{2_2}}, \ldots,$
21:        $C_{2,a_{2_t}} \ldots, C_{z,a_{z_1}}, C_{z,a_{z_2}}, \ldots, C_{z,a_{z_t}},$
22:        where $a_{s_k} \in \{i_{s_0}, i_{s_1}, \ldots, i_{s_t}\}$ and $a_{s_k} \ne a_{s_{k'}}$,
23:        $\forall k, k' \in [1, t]$ and $\forall s = [1, z]$ to be m^{th} user,
24:        calculate the set f_m as
25:        $f_m = C_{1,e_1} \cap C_{2,e_2} \cap \cdots \cap C_{z,e_z}$ where
26:        $e_s = \{i_{s_0}, i_{s_1}, \ldots, i_{s_t}\} \setminus \{a_{s_1}, a_{s_2}, \ldots, a_{s_t}\},$
27:        $\forall s \in [1, z]$. We have $|f_m| = \mu_z$.
28:        Let $f_m := \{y_{m,1}, y_{m,2}, \ldots, y_{m,\mu_z}\}$.
29:        Calculate f_m as above for all the $(t+1)^z$ users in
30:        $\mathcal{X}$.
31:        **Transmit:** Now do the following μ_z transmissions
32:

$$\underset{m \in \mathcal{X}}{\oplus} W_{d_m, y_{m,s}}, \ \forall s \in [\mu_z]$$

33:        Note that there are μ_z transmissions for $\mathcal{X}$.
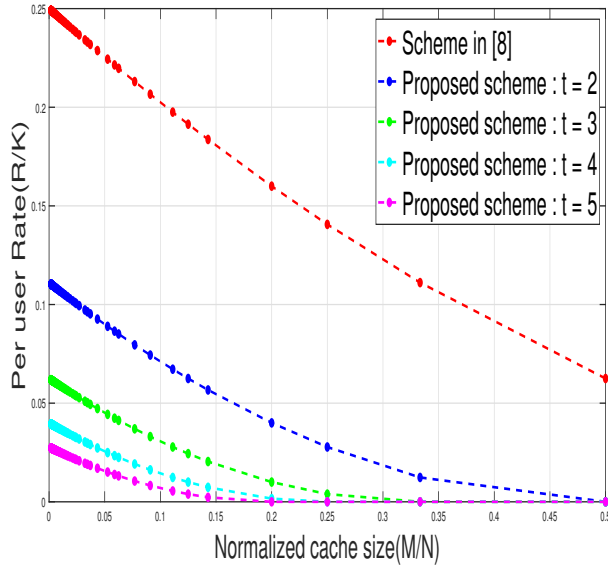34:   **end for**
35: **end for**

Figure 2. Comparison of the scheme in [12] and proposed scheme for the CRDs derived from affine resolvable BIBD's for the case $z = 2$, where q is a prime or prime power and $m \geq 2$.

III. PERFORMANCE COMPARISON

For performance comparison, we use the CRDs from affine resolvable designs obtained through affine geometry. Such CRDs exists for all q and m, where q is a prime or a power of a prime number and $m \geq 2$. The number of points $v = q^m$, the number of points in a block $k = q^{m-1}$, the number of blocks $b = \frac{q(q^m-1)}{q-1}$, the number of parallel classes $r = \frac{q^m-1}{q-1}$. It is known that any two blocks drawn from different parallel classes always intersect at exactly $\frac{k^2}{v}$ points [13] i.e., $z = 2$ and $\mu_z = q^{m-2}$.

1) Comparison with the scheme in [12]: For the multi-access coded caching scheme from CRDs with parameters q and m, we have, $K = (b_r)^z \binom{r}{z} = \frac{q(q^m-1)(q^{m-1}-1)\binom{q}{t}^2}{2(q-1)^2}$ users, $b = \frac{q(q^m-1)}{q-1}$ caches each having a cache size of $\frac{M}{N} = \frac{k}{v} = \frac{1}{q}$ and $\frac{M'}{N} = \frac{(2qt-t^2)}{q^2}$.

Fig 2 shows the variation of per user rate versus normalized cache size $\frac{M}{N}$. It can be seen that per-user-rate decreases as t increases.

Fig 3 shows how K, R and $\frac{R}{K}$ vary with respect to t. The case when $t = 1$, corresponds to the scheme in [12]. It is seen that the per user rate decreases as t increases. The number of users increase as t increases and then reaches a maximum after which it starts decreasing due to the binomial coefficient in the expression for K. Similar behavior can be seen for R also.

Fig. 4 depicts subpacketization (F) versus number of users for different values of t. It can be seen that for the same subpacketization levels, the number of users increases as t increases.
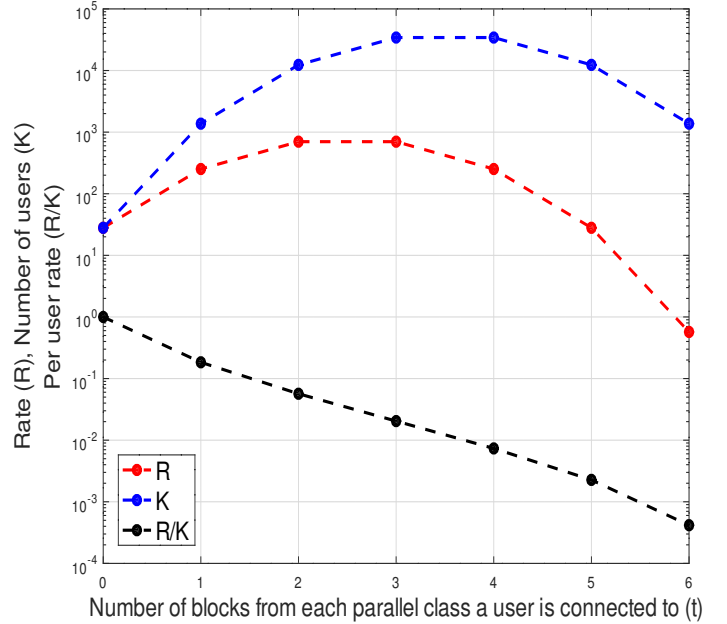

Figure 3. Comparison of Rate (R), Number of users (K) and Per user rate $\frac{R}{K}$ with respect to t for CRDs with parameters $v = 49$, $b = 56$, $k = 7$, $r = 8$, $z = 2$ and $\mu_2 = 1$
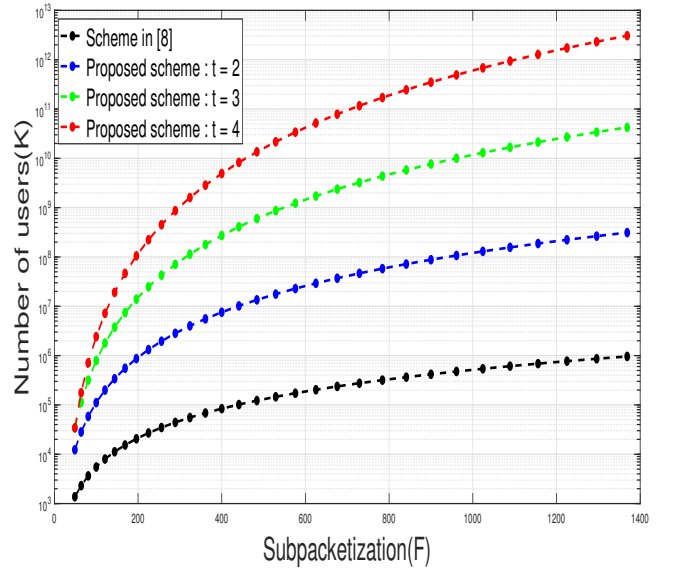

Figure 4. Comparison of subpacketization levels of the scheme in [12] and proposed scheme, $t \in \{2, 3, 4\}$ for the CRDs derived from affine resolvable BIBD's , where q is a prime or prime power and $m = 2$.

Table I
COMPARISON OF MAN SCHEME AND PROPOSED SCHEME FOR q BEING A PRIME OR PRIME POWER AND $m \geq 2$

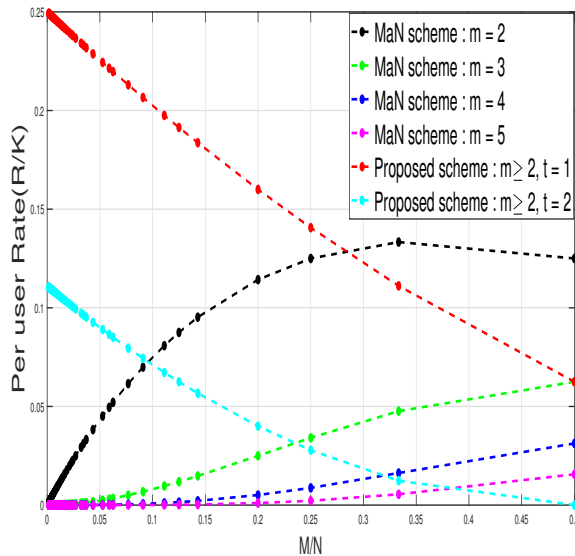| Parameters	MaN Scheme	Proposed Scheme
Number of Caches	$\dfrac{q(q^m - 1)}{q - 1}$	$\dfrac{q(q^m - 1)}{q - 1}$
Number of caches a user has access to	1	$2t$
Fraction of each file at each cache $\left(\frac{M}{N}\right)$	$\dfrac{1}{q}$	$\dfrac{1}{q}$
Number of Users (K)	$\dfrac{q(q^m - 1)}{q - 1}$	$\dfrac{q(q^m - 1)(q^{m-1} - 1)\binom{q}{t}^2}{2(q - 1)^2}$
Subpacketization level (F)	$\dbinom{q(q^m - 1)/q - 1}{(q^m - 1)/q - 1}$	q^m
Rate (R)	$\dfrac{(q^m - 1)(q - 1)}{q^m + q - 2}$	$\dfrac{(q^m - 1)(q^{m-1} - 1)\binom{q}{t+1}^2}{2q(q - 1)^2}$
Rate per user $\left(\frac{R}{K}\right)$	$\dfrac{(q - 1)^2}{q(q^m + q - 2)}$	$\left(\dfrac{\binom{q}{t+1}}{q\binom{q}{t}}\right)^2$
Gain (g)	$\dfrac{q^m - 1}{q - 1} + 1$	$(t + 1)^2$

Figure 5. Comparison between MaN and Proposed scheme for the CRDs derived from affine resolvable BIBD's for the case $z = 2$, where q is a prime or prime power and $m \geq 2$.

2) Comparison with the MaN scheme: Since $\frac{bM}{N} = \frac{b}{q} = \frac{(q^m - 1)}{q - 1}$ is a integer, there exist a coded caching system with users equipped with dedicated caches with $\frac{q(q^m - 1)}{q - 1}$ users and $\frac{M}{N} = \frac{1}{q}$. Table I shows the comparison of the MaN scheme and proposed scheme.

Fig. 5 shows the variation of per user rate $\frac{R}{K}$ versus the fraction of each file stored at each cache $\frac{M}{N}$. Since $\frac{M}{N} = \frac{1}{q}$ and $\frac{R}{K} = \frac{(q-1)^2}{(q)(q^m + q - 2)}$ in the case of the MaN scheme, and $\frac{R}{K} =$

$\left(\dfrac{\binom{q}{t+1}}{q\binom{q}{t}}\right)^2$ in the case of the proposed scheme is a function of q, we plot $\frac{R}{K}$ vs $\frac{M}{N}$ keeping m constant for different values of q, where q is a prime or prime power. From the plots, it can be concluded that by associating users to more blocks, advantages with respect to number of users supported, rate per user and subpacketization can be obtained.

IV. CONCLUSION

In [12], multi-access coded caching scheme is presented by taking one block from each of the parallel classes of the CRD used for the scheme. In this paper, it is shown that by allowing more than one block from each of the parallel classes lower per-user-rate can be obtained without increasing the subpacketization level.

ACKNOWLEDGMENT

REFERENCES

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

[2] 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.

[3] S. Agrawal, K. V. Sushena Sree and P. Krishnan, "Coded Caching based on Combinatorial Designs," in *2019 IEEE International Symposium on Information Theory (ISIT)*, Paris, France, 2019, pp. 1227-1231.

[4] 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.

[5] E. Parrinello, A. Unsal, and P. Elia, "Coded caching with shared caches: Fundamental limits with uncoded prefetching," *arXiv preprint arXiv:1809.09422*, 2018.

[6] A. M. Ibrahim, A. A. Zewail and A. Yener, "Coded Placement for Systems with Shared Caches," in *ICC 2019 - 2019 IEEE International Conference on Communications (ICC)*, Shanghai, China, 2019, pp. 1-6.

[7] B. Asadi and L. Ong, "Centralized Caching with Shared Caches in Heterogeneous Cellular Networks," in *2019 IEEE 20th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, Cannes, France, 2019, pp. 1-5.

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

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

[10] 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.

[11] M. Cheng, D. Liang, K. Wan, M. Zhang, G. Caire "A Novel Transformation Approach of Shared-link Coded Caching Schemes for Multiaccess Networks," *Available on arXiv:2012.04483 [cs.IT]*.

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

[13] D. R. Stinson, *Combinatorial Designs*: *Constructions and Analysis*. Springer, 2003.

[14] P. N. Muralidhar, D. Katyal and B. S. Rajan, "Improved Multi-access Coded Caching Schemes from Cross Resolvable Designs," *arXiv:2102.01372v2 [cs.IT]* (To appear in *2021 IEEE Information Theory Workshop (ITW2021), Kanazawa, Japan*, October 17-21, 2021).