Coded Caching Schemes for Multi-Access Topologies via Combinatorial Design Theory

Minquan Cheng, Kai Wan, Petros Elia, Giuseppe Caire

Coded Caching Schemes for Multi-Access Topologies via Combinatorial Design Theory

Minquan Cheng*, Kai Wan†, Petros Elia‡, Giuseppe Caire§
* Guangxi Normal University, 541004 Guilin China, chengqinshi@hotmail.com
† Huazhong University of Science and Technology, 430074 Wuhan, China, kai_wan@hust.edu.cn
‡ The Communication Systems Department, EURECOM, 06410 Sophia Antipolis, France, elia@eurecom.fr
§ Technische Universität Berlin, 10587 Berlin, Germany, caire@tu-berlin.de

***Abstract*—This paper studies a novel multi-access coded caching (MACC) model where the topology between users and cache nodes is a generalization of those already studied in previous work, such as combinatorial and cross-resolvable design topologies. Our goal is to minimize the worst-case transmission load in the delivery phase from the server over all possible user requests. By formulating the access topology as two classical combinatorial structures, t-design and t-group divisible design, we propose two classes of coded caching schemes for a flexible number of users, where the number of users can scale linearly, polynomially or exponentially with the number of cache nodes. In addition, our schemes can unify most schemes for the shared link network and unify many schemes for the multi-access network except for the cyclic wrap-around topology.**

***Index Terms*—Coded caching, multi-access networks, t-design, t-group divisible design**

I. Introduction

Caching can effectively shift traffic from peak to off-peak times [1] by storing fractions of popular content in users' local memories during peak traffic times, so that users can be partially served from their local caches, thereby reducing network traffic. The seminal work in [2] provided the so-called coded caching framework, which managed to achieve amazing coding gains that scale linearly with the total number of participating users for a shared-link network. The authors of [2] also proposed the first known scheme, referred to as the Maddah-Ali and Niesen (MN) scheme. From this work in [2], various works have been proposed for the original shared link network model and also as many extended models, where many of these works aimed at characterizing the optimal trade-off between memory size and communication load. While the mature literature on coded caching has explored a rich variety of settings, some recent findings - which we discuss later - have brought to the fore a powerful new way of exploiting a modest number of caches. This new way, called multi-access coded caching (MACC), is the subject of our work here. In this paper we consider this MACC model, where cache contents are stored at edge cache nodes in the network and users do not have their own caches, as shown in Fig. 1. The MACC model was originally proposed in [3] and it included a central server with N equal-length files, K cacheless users, and Γ cache nodes, where each user can access a subset of cache nodes at negligible cost. The MACC process consists of two phases, namely the placement phase and the delivery phase. (i) In the placement phase, each cache node places the equivalent of M files into its memory, aware of the network topology but without of knowledge of the users' subsequent demands. (ii) In the delivery phase, each user requests a signle file. All the requests are done simultaneously, and we assume here that each requested file is different than any other requested file. According to the users' demands and the contents cached by the cache nodes, the server broadcasts to the users the equivalent of R unites of files. The transmission is specifically designed such that each user can recover its demand from the broadcasted messages and the contents cached by its connected cache nodes. The objective is to characterise the trade-off between memory size M and the load R for the worst-case where all the demands are different.

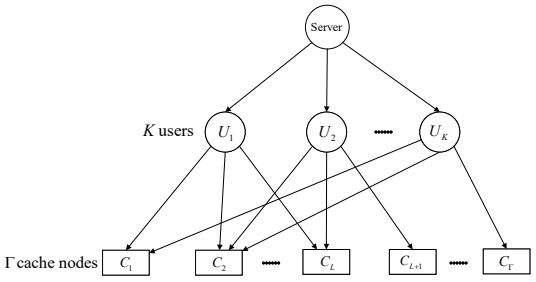

Fig. 1: Multiaccess coded caching system.

The MACC model was first considered with access topology in [3], where each user can access L neighbouring cache nodes in a cyclic wrap-around fashion. The authors in [3] proposed a coloring-based coded caching scheme that exploits the network topology. Under the one-dimensional (1D) MACC model, several improved schemes (with lower load or/and lower subpacketization) and converse bounds on the load were proposed in [4]–[14]. In addition, some works have extended 1D MACC to 2D MACC [13], [15]. Recently, [16] introduced a very powerful new MACC topology which proved to have stunning gains. In particular, [16] introduced the so called combinatorial access topology which included Γ cache nodes and $\binom{\Gamma}{L}$ users where each L subset of cache nodes is connected to a distinct user. A coded caching scheme was

proposed in [16], which was then proved in [17] to achieve optimal load under the constraint of uncoded cache placement where each cache node directly selects some library bits to store. Another very interesting access topology considered in the MACC literature is the cross-resolvable design in [11].

In fact the combinatorial access topology is equivalent to a special t-design with $t = L$, which consists of a Γ-point set $\mathcal{X}$ and family containing some subsets (referred to as blocks) of $\mathcal{X}$ (See Lemma 4 in Section II-C). The cross-resolvable design access topology is equivalent to a special $t = L$-GDD which consists of $\mathcal{X}$ and its partitions and a family of blocks (See Remark 2 in Section III). In addition, the number of users equals the number of blocks. Furthermore it is well known that by choosing the different values of t, the block numbers of these two classes can scale linearly, polynomially or exponentially with the number of cache nodes.

According to the property of the t-design, i.e., any t-subset of $\mathcal{X}$ is contained by a certain number of blocks, here we apply the placement strategy of the MN scheme into the t-design access topology, to obtain our scheme. Interestingly, our scheme also covers the scheme in [16] as a special case. Then according to the property of t-GDD, i.e., it has some partition of $\mathcal{X}$, by applying an orthogonal array (OA) placement strategy in the context of the t-GDD access topology, we obtain the second proposed scheme. As a by-product, it is also interesting to see that our schemes can also work and unify some existing schemes for the original MN shared-link caching network, where each user access its own (single) cache. Furthermore by carefully calibrating our OA and t-GDD structure, we can show that –with the exception of the cyclic-wrap-around topology–most other existing schemes and structures can be captured by our new unifying setting.

Notations: We assume that all sets are in an ordered set; for a set $\mathcal{V}$, we let $\mathcal{V}(j)$ represent the j-th element in the ordering of $\mathcal{V}$ and let $\mathcal{V}(\mathcal{J}) = \{\mathcal{V}(j)|j \in \mathcal{J}\}$. For any positive integers a, b, t with $a < b$ and $t \leq b$, and any non-negative set $\mathcal{V}$, let $[a, b] = \{a, a+1, \ldots, b\}$, especially $[1, b]$ be shorten by $[b]$, and $\binom{[b]}{t} = \{\mathcal{V} \mid \mathcal{V} \subseteq [b], |\mathcal{V}| = t\}$, i.e., $\binom{[b]}{t}$ is the collection of all t-sized subsets of $[b]$.

II. System Models

In this section, we introduce the multiaccess caching model studied in this paper. For the original MN caching model in [2], and the MN caching scheme from the viewpoint of placement delivery array (PDA) [18] are recalled. Then some related combinatorial concepts are reviewed.

A. The original MN caching system and MN PDA

In the shared-link coded caching system [2], a server containing N files with equal length in $\mathcal{W} = \{W_1, \ldots, W_N\}$ connects through an error-free shared link to K users in $\{U_1, U_2, \ldots, U_K\}$ with $K \leq N$, and every user has a cache which can store up to M files for $0 \leq M \leq N$. An F-division (K, M, N) coded caching scheme contains two phases.

• *Placement Phase.* Each file is divided into F packets with equal size,[1] and then each user U_k where $k \in [K]$ caches some packets of each file, which is limited by its cache size M. Let $\mathcal{Z}_{U_k}$ denote the cache contents at user U_k.

• *Delivery Phase.* Each user randomly requests one file from the server. The requested file by user U_k is represented by $W_{d_{U_k}}$, and the request vector by all users is denoted by $\mathbf{d} = (d_{U_1}, \ldots, d_{U_K})$. According to the cached contents and requests of all the users, the server transmits a broadcast message including $S_{\mathbf{d}}$ packets to all users, such that each user's request can be satisfied.

In such system, the number of worst-case transmitted files (a.k.a. load) over all possible requests is expected to be as small as possible, which is defined as $R = \max_{\mathbf{d} \in [N]^K} \frac{S_{\mathbf{d}}}{F}$.

For the shared-link coded caching problem, the authors in [18] proposed a class of solutions based on the concept of placement delivery array (PDA) which is defined as follows.

Definition 1: ([18]) For positive integers K, F, Z and S, an $F \times K$ array $\mathbf{P} = (p_{j,k})_{j \in [F], k \in [K]}$, composed of a specific symbol " $*$ " and S positive integers from $[S]$, is called a (K, F, Z, S) PDA if it satisfies the following conditions:

C1. The symbol " $*$ " appears Z times in each column;

C2. Each integer occurs at least once in the array;

C3. For any two distinct entries p_{j_1,k_1} and p_{j_2,k_2}, $p_{j_1,k_1} = p_{j_2,k_2} = s$ is an integer only if a.) $j_1 \neq j_2$, $k_1 \neq k_2$; and b.) $p_{j_1,k_2} = p_{j_2,k_1} = *$.

Lemma 1: ([18]) An F-division caching scheme for a (K, M, N) caching system can be realized by a (K, F, Z, S) PDA with $M/N = Z/F$ and load $R = S/F$.

Finally let us briefly introduce the MN coded caching scheme in [2] from the viewpoint of MN PDA, where the resulting PDA is referred to as MN PDA. For any integer $t \in [K]$, a $\binom{K}{t} \times K$ array $\mathbf{P} = (p_{\mathcal{T},k})$, $\mathcal{T} \in \binom{[K]}{t}$, $k \in [K]$, is defined by $p_{\mathcal{T},k} = \mathcal{T} \cup \{k\}$ if $k \notin \mathcal{T}$, otherwise $p_{\mathcal{T},k} = *$ where the rows are labelled by all the subsets $\mathcal{T} \in \binom{[K]}{t}$. When randomly choosing two bijections from the sets $\binom{[K]}{t}$ and $\binom{[K]}{t+1}$ to the $[\binom{K}{t}]$ and $[\binom{K}{t+1}]$ respectively, the following MN PDA and its achieved load can be obtained.

Lemma 2: (MN PDA [2]) For any positive integers K and t with $t < K$, there exists a $(K, \binom{K}{t}, \binom{K-1}{t-1}, \binom{K}{t+1})$ PDA, which gives a $\binom{K}{t}$-division (K, M, N) coded caching scheme for original MN caching model with $M/N = t/K$ and load $R = (K - t)/(t + 1)$.

B. Multiaccess caching system

The (L, r, K, Γ, M, N) multiaccess coded caching problem containing a server with a set of N equal-length files (denoted by $\mathcal{W} = \{W_1, \ldots, W_N\}$), Γ cache nodes (denoted by

[1] In this paper, we only consider the uncoded cache placement.

$C_1, \ldots, C_\Gamma$), and $K \leq N$ users (denoted by $U_1, \ldots, U_K$). Each cache-node has a memory size of M files where $0 \leq M \leq N/L$. For any integers $\gamma \in [\Gamma]$ and $k \in [K]$, let $\mathcal{C}_\gamma$ denote the user set each of which can access cache node C_γ and $\mathcal{B}_k$ denote the cache node set each of which can be accessed by user U_k. Assume that each cache node is accessed by a different user set of size r and each user can access a different cache node set of size L, i.e., both $\mathcal{C}_\gamma$ and $\mathcal{B}_k$ are unique sets and $|\mathcal{C}_\gamma| = r$ and $|\mathcal{B}_k| = L$ for each $\gamma \in [\Gamma]$ and $k \in [K]$. So we have $r\Gamma = LK$. We call the above model of a caching system as (L, r, K, Γ, M, N) coded caching problem under access topology $\mathfrak{B} = \{\mathcal{B}_k \mid k \in [K]\}$. An F-division (L, r, K, Γ, M, N) coded caching scheme under access topology $\mathfrak{B}$ runs in two phases:

• Placement phase: Each file is divided into F packets of equal size, and then each cache-node C_γ where $\gamma \in [\Gamma]$, directly caches some packets of each file, which is limited by its cache size M. Let $\mathcal{Z}_{C_\gamma}$ denote the cache contents at cache-node C_γ. The placement phase is also done without knowledge of later requests. Each user U_k where $k \in [K]$ can retrieve the packets cached at the L cache-nodes of $\mathcal{B}_k$. Let $\mathcal{Z}_{U_k}$ denote the retrievable packets by user U_k.

• Delivery phase: Each user randomly requests one file. According to the request vector $\mathbf{d} = (d_{U_1}, d_{U_2}, \ldots, d_{U_K})$, the cached packets in cache-nodes and access topology, the server transmits $S_\mathbf{d}$ coded packets to all users, such that each user's request can be satisfied.

We aim to design a multiaccess coded caching scheme with minimum worst-case load. For the sake of simplicity, we do not distinguish the user U_k and its accessible cache node set $\mathcal{B}_k$ unless otherwise stated. Then any fixed access topology can be represented by an appropriate combinatorial structure. In this paper we will discuss the heterogeneous access topologies from the view points of combinatorial design theory. In the following we will introduce some classic combinatorial concepts and their relationships.

C. t-design, GDD and OA

Definition 2: ([19], Design) A design is a pair $(\mathcal{X}, \mathfrak{B})$ such that the following properties are satisfied 1) $\mathcal{X}$ is a set of elements called points, and 2) $\mathfrak{B}$ is a collection of nonempty subsets of $\mathcal{X}$ called blocks.

Definition 3: ([20], t-design) Let Γ, K, L and t and λ be five positive integers. A t-(Γ, L, K, λ)-design is a design $(\mathcal{X}, \mathfrak{B})$ where $\mathcal{X}$ has Γ points and $\mathfrak{B}$ has K blocks that satisfy 1) $|\mathcal{B}| = L$ for any $\mathcal{B} \in \mathfrak{B}$, and 2) every t-subset of $\mathcal{X}$ is contained in exactly λ blocks.

From Definition 3, we can obtain that the number of blocks is $K = \lambda\binom{\Gamma}{t} \Big/ \binom{L}{t} \approx O\left(\Gamma^t\right)$ $(\Gamma \to \infty)$. In addition, each point appears the same time in all blocks, which is $r = KL/\Gamma = \lambda\binom{\Gamma-1}{t-1}/\binom{L-1}{t-1}$. So a t-(Γ, L, K, λ)-design is also shorten as t-(Γ, L, λ)-design and written as t-$(\Gamma, L, K, r, \lambda)$-design sometimes in this paper. A t-(Γ, L, λ)-design is also a t'-$(\Gamma, L, \lambda_{t'})$ design where $t' \leq t$ and $\lambda_{t'} = \lambda\binom{\Gamma-t'}{t-t'} \Big/ \binom{L-t'}{t-t'}$. The 2-$(\Gamma, L, \lambda)$ design is always called (Γ, L, λ) balanced incomplete block design (in short BIBD).

Example 1: When $\Gamma = 7$ and $L = 3$, let $\mathcal{X} = \{1, 2, 3, 4, 5, 6, 7\}$ and $\mathfrak{B} = \{\{1, 2, 4\}, \{2, 3, 5\}, \{3, 4, 6\}, \{4, 5, 7\}, \{5, 6, 1\}, \{6, 7, 2\}, \{7, 1, 3\}\}$. Then $(\mathcal{X}, \mathfrak{B})$ is a 2-$(7, 3, 1)$ design, i.e., $(7, 3, 1)$ BIBD by Definition 3.

Recently, P. Keevash in [21] and S. Glock et al., in [22] respectively prove the existence conjecture for t-design, i.e., the following result.

Lemma 3 (The existence conjecture for t-design [21], [22])**:** Given t, L and λ, there exists an integer $\Gamma_0(t, L, \lambda)$ (which is a function of (t, L, λ)) such that for any $\Gamma > \Gamma_0(t, L, \lambda)$, a t-(Γ, L, λ) design exists if and only if for any $0 \leq i \leq t-1$, $\binom{L-i}{t-i}$ divides $\lambda\binom{\Gamma-i}{t-i}$.

The following special design, i.e., group divisible design, is also useful in this paper.

Definition 4: ([20], t-GDD) Let L, t, q and m be positive integers with $t \leq L \leq m$. A (m, q, L, λ) *group divisible t-design* (t-(m, q, L, λ) GDD) is a triple $(\mathcal{X}, \mathfrak{G}, \mathfrak{B})$ where 1) $\mathcal{X}$ is a set of mq points, 2) $\mathfrak{G} = \{\mathcal{G}_1, \mathcal{G}_2, \ldots, \mathcal{G}_m\}$ is a partition of $\mathcal{X}$ into m subsets each of which has size q (called groups), and 3) $\mathfrak{B}$ is a family of L-blocks of $\mathcal{X}$ such that every block intersects every group in at most one point, and every t-subset of points from t distinct groups belongs to exactly λ blocks.

We can obtain the number of blocks in $\mathfrak{B}$ by Definition 4 $K = \lambda\binom{m}{t}q^t/\binom{L}{t}$. Clearly the number of blocks in a t-design is larger than that of a t-GDD for the same point set, block size and index λ.

Definition 5: ([11], Cross resolvable design) A design $(\mathcal{V}, \mathfrak{A})$ is called resolvable if the blocks of $\mathfrak{A}$ can be partitioned into parallel classes, say $\mathfrak{A}_1$, $\mathfrak{A}_2$, …, $\mathfrak{A}_m$, each of which is a set of blocks that partition the set of elements. $(\mathcal{V}, \mathfrak{A})$ is called t-cross resolvable if the intersection of any t blocks drawn from any L distinct parallel classes has the same size λ_t which is called t-th cross intersection number.

A t-cross resolvable design $(\mathcal{V}, \mathfrak{A})$ is written as t- $(v, k, \Gamma, m, \lambda_t)$ resolvable design when $|\mathcal{V}| = v$ and $\mathfrak{A}$ has Γ blocks each of which has size k and has m parallel classes.

For any design $(\mathcal{V}, \mathfrak{A})$, we regard the blocks $\mathfrak{A}$ as points and $\mathcal{V}$ as block set where $\mathcal{A} \in \mathfrak{A}$ is contained by $x \in \mathcal{V}$ if and only if $x \in \mathcal{A}$, then the obtained design $(\mathfrak{A}, \mathcal{V}) = (\mathcal{X}, \mathfrak{B})$ is called the dual design of $(\mathcal{V}, \mathfrak{A})$. Clearly a design is a dual design of its dual design.

Example 2: Let $\mathcal{V} = [4]$ and $\mathfrak{A} = \mathfrak{A}_1 \cup \mathfrak{A}_2 \cup \mathfrak{A}_3$ where $\mathfrak{A}_1 = \{\mathcal{A}_1 = \{1, 3\}, \mathcal{A}_2 = \{2, 4\}\}$, $\mathfrak{A}_2 = \{\mathcal{A}_3 = \{1, 2\}, \mathcal{A}_4 = \{3, 4\}\}$ and $\mathfrak{A}_3 = \{\mathcal{A}_5 = \{1, 4\}, \mathcal{A}_6 = \{2, 3\}\}$. It is easy to check that $\mathfrak{A}_1$, $\mathfrak{A}_2$ and $\mathfrak{A}_3$ are three different parallel classes. Furthermore the intersection of any two blocks from different parallel classes contains exactly one point. So by Definition 5, $(\mathcal{V}, \mathfrak{A})$ is a t-$(v, k, \Gamma, m, \lambda_t) = 2 =$

$(4,2,6,3,1)$ resolvable design. Then its dual design $(\mathcal{X},\mathfrak{B})$ where $\mathcal{X}=\mathfrak{A}$ and all the blocks are $\mathfrak{B}=\{\{\mathcal{A}_1,\mathcal{A}_3,\mathcal{A}_5\},\{\mathcal{A}_2,\mathcal{A}_3,\mathcal{A}_6\},\{\mathcal{A}_1,\mathcal{A}_4,\mathcal{A}_6\},\{\mathcal{A}_2,\mathcal{A}_4,\mathcal{A}_5\}\}$. Define $\mathfrak{G}=\{\mathcal{G}_1=\{1,2\},\mathcal{G}_2=\{3,4\},\mathcal{G}_3=\{5,6\}\}$. We can check that $(\mathcal{X},\mathfrak{G},\mathfrak{B})$ is a t-$(m,q,m,\lambda)=2$-$(3,2,3,1)$ GDD.

In fact any dual of a cross resolvable design is a GDD.

Lemma 4: The dual of a t-(v,k,Γ,m,λ_t) resolvable design $(\mathcal{V},\mathfrak{A})$ is a t-(m,q,m,λ_t) GDD where $q=v/k$.

Finally the following concept is useful for our placement strategy of our second class of schemes.

Definition 6: ([19], OA) Let $\mathbf{A}$ be an $F_1\times m$ matrix over $[q]$ for positive integers F_1, m, $q\geq 2$, and $s\leq m$. $\mathbf{A}$ is an orthogonal array (OA) with strength s, denoted by $\mathrm{OA}_\lambda(F_1,m,q,s)$, if every $1\times s$ row vector appears exactly λ times in $\mathbf{A}|_{\mathcal{S}}$ for each $\mathcal{S}\in\binom{[m]}{s}$, where $\mathbf{A}|_{\mathcal{S}}$ represents the column-wise sub-matrix of $\mathbf{A}$ including the columns with indices in $\mathcal{S}$.

It is well known that $F_1=\lambda q^s$ for any $\mathrm{OA}_\lambda(F_1,m,q,s)$ and thus $\mathrm{OA}_\lambda(F_1,m,q,s)$ is sometimes written as $\mathrm{OA}_\lambda(m,q,s)$ for short [19]. The parameter λ is the index of the orthogonal array. If λ is omitted, then it is understood to be 1.

III. Main results

Firstly, by the MN placement strategy and t-design access topology, we can obtain the following result.

Theorem 1 (Scheme via t-design)**:** Assume that there exists a t-$(\Gamma,L,1)$ design $(\mathcal{X},\mathfrak{B})$ for some positive integers Γ, L and $t\geq 2$. For any positive integer M and N satisfying that $\Gamma M/N$ is an integer, we can obtain a $(L,r=\binom{\Gamma-1}{t-1}/\binom{L-1}{t-1},K=\binom{\Gamma}{t}/\binom{L}{t},\Gamma,M,N)$ coded caching scheme under access topology $\mathfrak{B}$ with memory ratio $\mu=M/N$, subpacketization $F=\binom{\Gamma}{\mu\Gamma}\binom{L}{t}$ and transmission load

$$R\leq\frac{\binom{\Gamma}{t+\mu\Gamma}}{\binom{\Gamma}{\mu\Gamma}\binom{L}{t}}-\frac{\sum_{i=1}^{\mu\Gamma-1}\binom{L}{t+i}\binom{\Gamma-L}{\mu\Gamma-i}}{\binom{L}{t}\binom{\Gamma}{\mu\Gamma}}-\frac{K\binom{L}{t+\mu\Gamma}}{\binom{L}{t}\binom{\Gamma}{\mu\Gamma}}\qquad(1)$$

□

By Lemma 3 and Theorem 1, we can obtain arbitrary t-(Γ,L,λ) designs for any parameters t, Γ, L and λ when Γ is large. Then by Theorem 1, we can obtain the $(L,r=\binom{\Gamma-1}{t-1}/\binom{L-1}{t-1},K=\binom{\Gamma}{t}/\binom{L}{t},\Gamma,M,N)$ multiaccess coded caching scheme with memory ratio $\mu=M/N$, subpacketization $F=\binom{\Gamma}{\mu\Gamma}\binom{L}{t}$ and transmission load in (1) for any parameters t, Γ, L and λ when Γ is large. So the following investigation can be obtained.

Remark 1 (Flexible number of users)**:** A careful observation of the number of blocks in a t-design reveals how our scheme can be nicely calibrated to fit a broad range of possible number of users. As one can readily conclude, this number of users can scale both linearly, polynomially as well as exponentially in the number of cache nodes Γ.

In addition, for any positive integers Γ and L there always exists a L-$(\Gamma,L,K=\binom{\Gamma}{L},r=\binom{\Gamma-1}{L-1},1)$ design $(\mathcal{X}=[\Gamma],\mathfrak{B}=\binom{[\Gamma]}{L})$. Then the following statement can be obtained.

Remark 2: The combinatorial access topology proposed in [23] is exactly the L-$(\Gamma,L,K=\binom{\Gamma}{L},r=\binom{\Gamma-1}{L-1},1)$ design access topology. Then the obtained $\binom{\Gamma}{\mu\Gamma}\times\binom{\Gamma}{L}$ array $\mathbf{Q}$ in $(\binom{\Gamma}{L},\binom{\Gamma}{\mu\Gamma},\binom{\Gamma}{\mu\Gamma}-\binom{\Gamma-L}{\mu\Gamma},\binom{\Gamma}{\mu\Gamma+L})$ PDA which leads to a $(L,r=\binom{\Gamma-1}{L-1},K=\binom{\Gamma}{L},M,N)$ multiaccess coded caching scheme based on combinatorial access topology with memory ration $\mu=\frac{M}{N}$, subpacketization $F=\binom{\Gamma}{\mu\Gamma}$ and load $R=\binom{\Gamma}{\mu\Gamma+L}/\binom{\Gamma}{\mu\Gamma}$. This scheme is exactly the scheme proposed in [23], which was shown to be optimal under uncoded cache placement and combinatorial access topology [17].

Secondly, using the OA placement strategy and t-GDD access topology, we obtain the following result.

Theorem 2 (Scheme via t-GDD)**:** For any positive integers m, q, L, t and s with $1\leq t\leq L\leq s\leq m$, if there exists a t-$(m,q,L,1)$ GDD and $\mathrm{OA}(m,q,s)$, then the following multiaccess schemes can be obtained.

- When $L=t$ and $s=m-1$, there exists a $(\Gamma=mq,K=\binom{m}{t}q^t,M,N)$ coded caching scheme under the t-$(m,q,t,1)$ GDD and $\mathrm{OA}(m,q,m-1)$ placement with $\frac{M}{N}=\frac{1}{q}$, subpacketization $F=q^{m-1}$ and transmission load $R=(q-1)^t$;
- When $L=t$ and $s+t=m$, there exists a $(\Gamma=mq,K=\binom{m}{t}q^t,M,N)$ coded caching scheme under the t-$(m,q,t,1)$ GDD and $\mathrm{OA}(m,q,s)$ placement with $\frac{M}{N}=\frac{1}{q}$, subpacketization $F=q^s$ and transmission load $R=q^t-1$;
- When $s=m$, there exists a $(\Gamma=mq,K=\binom{m}{t}q^t/\binom{L}{t},M,N)$ coded caching scheme under the t-$(m,q,L,1)$ GDD and $\mathrm{OA}(m,q,m)$ placement with $\frac{M}{N}=\frac{1}{q}$, subpacketization $F=q^m\binom{L}{t}$ and transmission load $R=(q-1)^t/\binom{L}{t}$.

Finally it is worth noting that similarly we can also use t-(Γ,L,λ) design and t-(m,q,L,λ) GDD access topology to construct the multiaccess coded caching schemes. Due to the space of this paper we omit it.

IV. Sketch of the proposed scheme in Theorem 1

Let us consider an $(L,r,K,\Gamma,M,N)=(3,3,7,7,1,7)$ coded caching scheme under the t-$(\Gamma,L,K,\lambda)=2$-$(7,3,7,1)$ design $(\mathcal{X},\mathfrak{B})$ (in Example 1) access topology. Then we have $\mu\Gamma=1$ and $\binom{L}{t}=\binom{3}{2}=3$.

By using the MN scheme and the 2-$(7,3,7,1)$ design, we first divide each file into 7 packets with equal size, i.e., for each $n\in[7]$, $W_n=(W_{n,\mathcal{D}})_{\mathcal{D}\in\binom{[7]}{1}}$, and further divide each packet into 3 subpackets with equal size, i.e., for each $n\in[N]$ and $\mathcal{D}\in\binom{[7]}{1}$, $W_{n,\mathcal{D}}=(W^{\mathcal{T}}_{n,\mathcal{D}})_{\mathcal{T}\in\binom{[3]}{2}}$. Let us introduce our main idea by constructing three arrays $\mathbf{C}$, $\mathbf{U}$, and $\mathbf{Q}$.

Definition 7 ([4])**:** • An $\binom{\Gamma}{\mu\Gamma}\binom{L}{t}\times K$ node-placement array $\mathbf{C}$ consists of star and null, where F and K represent

TABLE I: Three arrays

(a) Node-placement array $\mathbf{C}$.

Subpackets	Cache node set $\mathcal{X}$						
$\mathcal{D},\mathcal{T}$	C_1	C_2	C_3	C_4	C_5	C_6	C_7
$\{1\},\{1,2\}$	*						
$\{2\},\{1,2\}$		*					
$\{3\},\{1,2\}$			*				
$\{4\},\{1,2\}$				*			
$\{5\},\{1,2\}$					*		
$\{6\},\{1,2\}$						*	
$\{7\},\{1,2\}$							*
$\{1\},\{1,3\}$	*						
$\{2\},\{1,3\}$		*					
$\{3\},\{1,3\}$			*				
$\{4\},\{1,3\}$				*			
$\{5\},\{1,3\}$					*		
$\{6\},\{1,3\}$						*	
$\{7\},\{1,3\}$							*
$\{1\},\{2,3\}$	*						
$\{2\},\{2,3\}$		*					
$\{3\},\{2,3\}$			*				
$\{4\},\{2,3\}$				*			
$\{5\},\{2,3\}$					*		
$\{6\},\{2,3\}$						*	
$\{7\},\{2,3\}$							*

(b) User-retrieve array $\mathbf{U}$.

Subpackets	User set $\mathfrak{B}$						
$\mathcal{D},\mathcal{T}$	U_{124}	U_{235}	U_{346}	U_{457}	U_{561}	U_{672}	U_{713}
$\{1\},\{1,2\}$	*				*		*
$\{2\},\{1,2\}$	*	*				*	
$\{3\},\{1,2\}$		*	*				*
$\{4\},\{1,2\}$	*		*	*			
$\{5\},\{1,2\}$		*		*	*		
$\{6\},\{1,2\}$			*		*	*	
$\{7\},\{1,2\}$				*		*	*
$\{1\},\{1,3\}$	*				*		*
$\{2\},\{1,3\}$	*	*				*	
$\{3\},\{1,3\}$		*	*				*
$\{4\},\{1,3\}$	*		*	*			
$\{5\},\{1,3\}$		*		*	*		
$\{6\},\{1,3\}$			*		*	*	
$\{7\},\{1,3\}$				*		*	*
$\{1\},\{2,3\}$	*				*		*
$\{2\},\{2,3\}$	*	*				*	
$\{3\},\{2,3\}$		*	*				*
$\{4\},\{2,3\}$	*		*	*			
$\{5\},\{2,3\}$		*		*	*		
$\{6\},\{2,3\}$			*		*	*	
$\{7\},\{2,3\}$				*		*	*

(c) User-retrieve array $\mathbf{Q}$.

Subpackets	User set $\mathfrak{B}$						
$\mathcal{D},\mathcal{T}$	U_{124}	U_{235}	U_{346}	U_{457}	U_{561}	U_{672}	U_{713}
$\{1\},\{1,2\}$	*	123	134	145	*	167	*
$\{2\},\{1,2\}$	*	*	234	245	256	*	127
$\{3\},\{1,2\}$	123	*	*	345	356	367	*
$\{4\},\{1,2\}$	*	234	*	*	456	467	147
$\{5\},\{1,2\}$	125	*	345	*	*	567	157
$\{6\},\{1,2\}$	126	236	*	456	*	*	167
$\{7\},\{1,2\}$	127	237	347	*	567	*	*
$\{1\},\{1,3\}$	*	125	136	147	*	126	*
$\{2\},\{1,3\}$	*	*	236	247	125	*	237
$\{3\},\{1,3\}$	134	*	*	347	135	236	*
$\{4\},\{1,3\}$	*	245	*	*	145	246	347
$\{5\},\{1,3\}$	145	*	356	*	*	256	357
$\{6\},\{1,3\}$	146	256	*	467	*	*	367
$\{7\},\{1,3\}$	147	257	367	*	157	*	*
$\{1\},\{2,3\}$	*	135	146	157	*	127	*
$\{2\},\{2,3\}$	*	*	246	257	126	*	123
$\{3\},\{2,3\}$	234	*	*	357	136	237	*
$\{4\},\{2,3\}$	*	345	*	*	146	247	134
$\{5\},\{2,3\}$	245	*	456	*	*	257	135
$\{6\},\{2,3\}$	246	356	*	567	*	*	136
$\{7\},\{2,3\}$	247	357	467	*	167	*	*

the subpacketization of each subfile and the number of cache-nodes, respectively. The entry located at the position (j,k) in $\mathbf{C}$ is star if and only if the k^{th} cache-node caches the j^{th} packet of each W_n where $n\in[N]$.

• An $\binom{\Gamma}{\mu\Gamma}\binom{L}{t}\times K$ user-retrieve array $\mathbf{U}$ consists of star and null, where F and K represent the subpacketization of each subfile and the number of users, respectively. The entry at the position (j,k) in $\mathbf{U}$ is star if and only if the k^{th} user can retrieve the j^{th} packet of each W_n where $n\in[N]$.

• An $\binom{\Gamma}{\mu\Gamma}\binom{L}{t}\times K$ user-delivery array $\mathbf{Q}$ consists of $\{*\}\cup[S]$, where F, K and the stars in $\mathbf{Q}$ have the same meaning as F, K of $\mathbf{U}$ and the stars in $\mathbf{U}$, respectively. Each integer represents a multicast message, and S represents the total number of transmitted multicast messages. □

1) Construction of node-placement array $\mathbf{C}$*:* By the placement strategy MN scheme, the packets cached by cache nodes can be written as $\mathcal{Z}_{C_1}=\{W_{n,\{1\}}^{\{1,2\}},W_{n,\{1\}}^{\{1,3\}},W_{n,\{1\}}^{\{2,3\}}|n\in[7]\}$, $\mathcal{Z}_{C_2}=\{W_{n,\{2\}}^{\{1,2\}},W_{n,\{2\}}^{\{1,3\}},W_{n,\{2\}}^{\{2,3\}}|n\in[7]\}$, $\mathcal{Z}_{C_3}=\{W_{n,\{3\}}^{\{1,2\}},W_{n,\{3\}}^{\{1,3\}},W_{n,\{3\}}^{\{2,3\}}|n\in[7]\}$, $\mathcal{Z}_{C_4}=\{W_{n,\{4\}}^{\{1,2\}},W_{n,\{4\}}^{\{1,3\}},W_{n,\{4\}}^{\{2,3\}}|n\in[7]\}$, $\mathcal{Z}_{C_5}=\{W_{n,\{5\}}^{\{1,2\}},W_{n,\{5\}}^{\{1,3\}},W_{n,\{5\}}^{\{2,3\}}|n\in[7]\}$, $\mathcal{Z}_{C_6}=\{W_{n,\{6\}}^{\{1,2\}},W_{n,\{6\}}^{\{1,3\}},W_{n,\{6\}}^{\{2,3\}}|n\in[7]\}$ and $\mathcal{Z}_{C_7}=\{W_{n,\{7\}}^{\{1,2\}},W_{n,\{7\}}^{\{1,3\}},W_{n,\{7\}}^{\{2,3\}}|\ n\in[7]\}$. We can see that each cache node caches $7\times 3=21$ subpackets, i.e., one file. By Definition 7, a 21×7 node-placement array $\mathbf{C}$ which represents the subpackets cached by the cache nodes is given in Table Ia.

2) Construction of user-retrieve array $\mathbf{U}$*:* By the 2-$(7,3,7,1)$ design $(\mathcal{X},\mathfrak{B})$ where $\mathfrak{B}=\{\{1,2,4\},\{2,3,5\},\{3,4,6\},\{4,5,6\},\{5,6,1\},\{6,7,2\},\{7,1,3\}\}$, the users $U_{\mathcal{B}}$ where $\mathcal{B}\in\mathfrak{B}$ can retrieve the following subpackets,

$\mathcal{Z}_{U_{124}}=\{W_{n,\{1\}}^{\{1,2\}},W_{n,\{1\}}^{\{1,3\}},W_{n,\{1\}}^{\{2,3\}},W_{n,\{2\}}^{\{1,2\}},W_{n,\{2\}}^{\{1,3\}},W_{n,\{2\}}^{\{2,3\}},$
$W_{n,\{4\}}^{\{1,2\}},W_{n,\{4\}}^{\{1,3\}},W_{n,\{4\}}^{\{2,3\}}|n\in[7]\}$,
$\mathcal{Z}_{U_{235}}=\{W_{n,\{2\}}^{\{1,2\}},W_{n,\{2\}}^{\{1,3\}},W_{n,\{2\}}^{\{2,3\}},W_{n,\{3\}}^{\{1,2\}},W_{n,\{3\}}^{\{1,3\}},W_{n,\{3\}}^{\{2,3\}},$
$W_{n,\{5\}}^{\{1,2\}},W_{n,\{5\}}^{\{1,3\}},W_{n,\{5\}}^{\{2,3\}}|n\in[7]\}$,
$\mathcal{Z}_{U_{346}}=\{W_{n,\{3\}}^{\{1,2\}},W_{n,\{3\}}^{\{1,3\}},W_{n,\{3\}}^{\{2,3\}},W_{n,\{4\}}^{\{1,2\}},W_{n,\{4\}}^{\{1,3\}},$
$W_{n,\{4\}}^{\{2,3\}},W_{n,\{6\}}^{\{1,2\}},W_{n,\{6\}}^{\{1,3\}},W_{n,\{6\}}^{\{2,3\}}|n\in[7]\}$
$\mathcal{Z}_{U_{457}}=\{W_{n,\{4\}}^{\{1,2\}},W_{n,\{4\}}^{\{1,3\}},W_{n,\{4\}}^{\{2,3\}},W_{n,\{5\}}^{\{1,2\}},W_{n,\{5\}}^{\{1,3\}},W_{n,\{5\}}^{\{2,3\}},$
$W_{n,\{7\}}^{\{1,2\}},W_{n,\{7\}}^{\{1,3\}},W_{n,\{7\}}^{\{2,3\}}|n\in[7]\}$,
$\mathcal{Z}_{U_{561}}=\{W_{n,\{5\}}^{\{1,2\}},W_{n,\{5\}}^{\{1,3\}},W_{n,\{5\}}^{\{2,3\}},W_{n,\{6\}}^{\{1,2\}},W_{n,\{6\}}^{\{1,3\}},$
$W_{n,\{6\}}^{\{2,3\}},W_{n,\{1\}}^{\{1,2\}},W_{n,\{1\}}^{\{1,3\}},W_{n,\{1\}}^{\{2,3\}}|n\in[7]\}$,
$\mathcal{Z}_{U_{672}}=\{W_{n,\{6\}}^{\{1,2\}},W_{n,\{6\}}^{\{1,3\}},W_{n,\{6\}}^{\{2,3\}},W_{n,\{7\}}^{\{1,2\}},W_{n,\{7\}}^{\{1,3\}},W_{n,\{7\}}^{\{2,3\}},$
$W_{n,\{2\}}^{\{1,2\}},W_{n,\{2\}}^{\{1,3\}},W_{n,\{2\}}^{\{2,3\}}|n\in[7]\}$,
$\mathcal{Z}_{U_{713}}=\{W_{n,\{7\}}^{\{1,2\}},W_{n,\{7\}}^{\{1,3\}},W_{n,\{7\}}^{\{2,3\}},W_{n,\{1\}}^{\{1,2\}},W_{n,\{1\}}^{\{1,3\}},W_{n,\{1\}}^{\{2,3\}},$
$W_{n,\{3\}}^{\{1,2\}},W_{n,\{3\}}^{\{1,3\}},W_{n,\{3\}}^{\{2,3\}}|n\in[7]\}$.

By Definition 7, we have a 21×7 user-retrieve array $\mathbf{U}$ to represent the packets retrieved by the users in Table Ia.

3) Construction of user-delivery array $\mathbf{Q}$*:* For each row label $(\mathcal{D},\mathcal{T})$ and column label $\mathcal{B}$, if $\mathcal{D}\cap\mathcal{B}=\emptyset$ we put the subset $\mathcal{D}\cup\mathcal{B}(\mathcal{T})$ into the entry $\mathbf{U}((\mathcal{D},\mathcal{T}),\mathcal{B})$ to the user-delivery array $\mathbf{Q}$. So in this example we can obtain the 21×7 user-delivery array $\mathbf{Q}$ which is listed in Table Ic. We can check that there are exactly $S=\binom{\Gamma}{t+\mu\Gamma}-K\binom{L}{t+\mu\Gamma}=\binom{7}{2+1}-7\times\binom{3}{2+1}=35-7=28$, and the obtained $\mathbf{Q}$ is a $(7,21,9,28)$ PDA which leads to a $(7,M',N)$ shared link coded caching scheme with $M'/N=3/7$, subpacketization 21 and transmission load $28/21=4/3$ by Lemma 1. The overall coded caching gain is $K(F-Z)/S=7\times4\times3/28=3$.

Acknowledgement: This work was partially funded by NSFC (62061004, U21A20474, 12141107), 2022GXNSFDA035087, the European Research Council under the ERC Advanced Grant N. 789190, CARENET, and the ERC Project DUALITY under Grant 725929.

REFERENCES

[1] E. Bastug, M. Bennis, and M. Debbah, "Living on the edge: The role of proactive caching in 5G wireless networks," *IEEE Communications Magazine*, vol. 52, no. 8, pp. 82–89, 2014.

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

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

[4] M. Cheng, K. Wan, D. Liang, M. Zhang, and G. Caire, "A novel transformation approach of shared-link coded caching schemes for multiaccess networks," *IEEE Transactions on Communications*, vol. 69, no. 11, pp. 7376–7389, 2021.

[5] S. Sasi and B. Sundar Rajan, "Multi-access coded caching scheme with linear sub-packetization using PDAs," *IEEE Transactions on Communications*, pp. 1–1, 2021.

[6] A. A. Mahesh and B. Sundar Rajan, "A coded caching scheme with linear sub-packetization and its application to multi-access coded caching," *arXiv:2009.10923*, 2020.

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

[8] K. S. Reddy and N. Karamchandani, "Rate-memory trade-off for multi-access coded caching with uncoded placement," *IEEE Transactions on Communications*, vol. 68, no. 6, pp. 3261–3274, 2020.

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

[10] K. S. Reddy and N. Karamchandani, "Structured index coding problem and multi-access coded caching," *IEEE Journal on Selected Areas in Information Theory*, vol. 2, no. 4, pp. 1266–1281, 2021.

[11] D. Katyal, P. N. Muralidhar, and B. S. Rajan, "Multi-access coded caching schemes from cross resolvable designs," *IEEE Transactions on Communications*, vol. 69, no. 5, pp. 2997–3010, 2021.

[12] K. Wan, M. Cheng, D. Liang, and G. Caire, "Multiaccess coded caching with private demands," in *2022 IEEE International Symposium on Information Theory (ISIT)*. IEEE, 2022, pp. 1390–1395.

[13] M. Zhang, K. Wan, M. Cheng, and G. Caire, "Coded caching for two-dimensional multi-access networks," *arXiv:2201.11465*, 2022.

[14] K. Wan, M. Cheng, M. Kobayashi, and G. Caire, "On the optimal memory-load tradeoff of coded caching for location-based content," *IEEE Transactions on Communications*, vol. 70, no. 5, pp. 3047–3062, 2022.

[15] E. Ozfatura and D. Gündüz, "Mobility-aware coded storage and delivery," *IEEE Transactions on Communications*, vol. 68, no. 6, pp. 3275–3285, 2020.

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

[17] F. Brunero and P. Elia, "Fundamental limits of combinatorial multi-access caching," *IEEE Transactions on Information Theory*, vol. 69, no. 2, pp. 1037–1056, 2023.

[18] Q. Yan, M. Cheng, X. Tang, and Q. Chen, "On the placement delivery array design for centralized coded caching scheme," *IEEE Transactions on Information Theory*, vol. 63, no. 9, pp. 5821–5833, 2017.

[19] D. R. Stinson, *Combinatorial designs: constructions and analysis*. Springer, NY, 2004.

[20] C. Colbourn and J. Dinitz, "Handbook of combinatorial designs," in *Chapman*, Hall/CRC, 2006, vol. 42.

[21] P. Keevash, "The existence of designs," *arXiv: 1401.3665*, Jan. 2014.

[22] A. L. Stefan Glock, Daniela Kuhn and D. Osthus, "The existence of designs via iterative absorption," *arXiv: 1611.06827*, Feb. 2016.

[23] P. N. Muralidhar, D. Katyal, and B. S. Rajan, "Maddah-ali-niesen scheme for multi-access coded caching," *IEEE Information Theory Workshop (ITW)*, 2021.