Coded Caching Design for D2D Networks With Reduced Subpacketizations

Xianzhang Wu, *Senior Member, IEEE*, Minquan Cheng, *Member, IEEE*, Li Chen, *Senior Member, IEEE*, Congduan Li, *Senior Member, IEEE*, Shuwu Chen, and Rongteng Wu, *Senior Member, IEEE*

***Abstract*—Device-to-Device (D2D) assisted coded caching is a promising approach to improve the communication efficiency over networks. However, the basic D2D coded caching scheme requires a subpacketization size that increases exponentially with the number of users. This is infeasible since the file size needs to be extremely large in the server. It is desirable to design a scheme that achieves a small subpacketization size while keeping the rate low. Recently, D2D placement delivery array (DPDA) was proposed to address the high subpacketization issue of D2D coded caching. This paper investigates the design of DPDA from the perspectives of linear algebraic and additive combinatorics. It is shown that a linear subspace possessing certain property can be employed in the design of DPDA. Based on this, a new D2D coded caching scheme with a subquadratic subpacketization size is derived through shortening the binary Reed-Muller codes. In order to obtain a D2D coded caching scheme with a linear subpacketization size, a new combinatorial structure called proper disjoint 3-term arithmetic progression (3-AP) free set is further introduced, and a deterministic algorithm for constructing it is provided with a polynomial complexity. Both the theoretical and numerical results reveal that the proposed schemes have a superior performance in terms of subpacketization size or transmission rate.**

***Index Terms*—Coded caching, placement delivery array, linear subspace, subpacketization size, proper disjoint 3-AP free set.**

Received 19 September 2025; revised 11 February 2026 and 12 March 2026; accepted 13 March 2026. Date of publication 24 March 2026; date of current version 3 April 2026. This work is sponsored by the Educational and Scientific Research Project for Young Teachers of Fujian Province under Grant KLY24207XA, the Guangxi Natural Science Foundation under Grant DA035087, the National Natural Science Foundation of China (NSFC) under Grants 62471503 and 62271514, the Project of Industry-university-institute Cooperation in Colleges and Universities in Fujian Province under Grant 2024H6007, and the Environmental Protection Science and Technology Planning Project of Fujian Province under Grant 2025R016. The associate editor coordinating the review of this article and approving it for publication was N. Akar. *(Corresponding authors: Li Chen; Congduan Li.)*

Xianzhang Wu and Shuwu Chen are with the College of Computer and Information Science, Fujian Agriculture and Forestry University, Fuzhou 350002, China, and also with the Engineering Research Center of Smart Sensing and Agricultural Chip Technology, Fujian Province University, Fuzhou 350002, China (e-mail: wuxianzhangll@163.com; chenshuwu@fafu.edu.cn).

Minquan Cheng is with the Key Laboratory of Education Blockchain and Intelligent Technology, Ministry of Education, and Guangxi Key Laboratory of Multi-Source Information Mining and Security, Guangxi Normal University, Guilin 541004, China (e-mail: chengqinshi@hotmail.com).

Li Chen is with the School of Electronics and Information Technology, Sun Yat-sen University, Guangzhou 510006, China (e-mail: chenli55@mail.sysu.edu.cn).

Congduan Li is with the School of Electronics and Communication Engineering, Sun Yat-sen University, Shenzhen 518107, China (e-mail: licongd@mail.sysu.edu.cn).

Rongteng Wu is with the School of Computer and Big Data, Minjiang University, Fuzhou 350108, China (e-mail: rongtengwu@163.com).

Digital Object Identifier 10.1109/TCOMM.2026.3677079

I. Introduction

The growing number of network users and their increasing appetite for video streaming services lead to a severe network congestion during the peak traffic times. Caching that exploits users' local memories to provide a faster demand service becomes a promising technology for future communication systems. A more effective way of caching is through coding, which was first studied by Maddah-Ali and Niesen (MN) [1]. It further reduces the network burden by taking advantage of the potential coded multicasting opportunities distributed over the networks. The whole procedure in coded caching typically involves two phases that are jointly designed to pursue a low communication load. In the placement phase, the server prefetches some proper contents to fill each user's cache without the knowledge of later demands. In the delivery phase, the server will be informed with the users' requests and then broadcasts some coded packets demanded by the users based on their previous cache contents. The worst case bits of the delivering packets normalized by the file size is called the transmission rate. Within this paradigm, if each user directly caches a subset of the packets without coding, it is called an uncoded placement. Otherwise, the placement is called coded placement.

Inspired by the seminal work of [1], several follow-up researches focused on the improvements of the achievable rates and lower bounds [2], [3], [4], [5]. Despite the significant benefits of coded caching for wireless networks, it suffers a practical challenge, i.e., each library file needs to be partitioned into F non-overlapping packets, where F is called subpacketization size. It increases exponentially with the number of users. There has been a large amount of work aiming to reduce the subpacketization size. But they are at the cost of slightly increasing the transmission rate [6], [7], [8], [9], [10], [11], [12], [13], [14], [15]. In particular, the work of [6] introduced a combinatorial array called placement delivery array (PDA) in realizing a coded caching scheme, and thereby constructed two new coded caching schemes that have a smaller subpacketization size than that of the MN scheme. By using the lifting construction for low-density parity-check codes, the work of [9] proposed some new coded caching schemes with a linear subpacketization size for large memory ratios. Based on the Hamming distance, the work of [12] constructed some new coded caching schemes that can yield a linear subpacketization size and a large coding gain for the coded placement setting. However, they can only support the scenario with a fixed number of users. Later, the work of [14] established a relationship between the injective arc coloring of regular digraphs and PDA. It can accommodate a flexible number of users and a linear subpacketization size. This connection further generalizes the one characterized by Ruzsa-Szemérédi graphs [15]. Other effective approaches for reducing the subpacketization size include the use of projective geometry and line graphs [7], hypergraphs [10], strong edge coloring of bipartite graph [11], and combinatorial design [13].

Recently, the MN scheme has been extended for the wireless D2D caching networks. It enables the users to communicate with each other for improving the energy and spectral efficiency, providing a stable communication environment with controllable interference. Hence, the study of coded caching for D2D networks becomes an active research topic [16], [17], [18], [19], [20], [21], [22], [23], [24], [25], [26], [27], [28], [29], [30]. For instance, the work of [16] proposed a D2D coded caching scheme that has an asymptotically optimal transmission rate for large memory ratios. In [17], an optimal rate-memory trade off of D2D coded caching was derived under the assumptions of uncoded placement and one-shot delivery. The work of [18] considered a D2D caching network scenario in which only a part of users join in delivering the missing packets to other network users. The scheme of [19] extended the work of [16] to a more general framework by introducing a new combinatorial array called D2D placement delivery array (DPDA). Similar to [19], the authors in [20] proposed a hypercube based D2D coded caching scheme with a subexponential subpacketization size. Based on the grouping method, a D2D coded caching scheme that achieves an optimal transmission rate with a smaller subpacketization size was proposed in [21]. In addition, the work of [22] proposed some new DPDA based coded caching schemes through modifying the existing PDAs. A more recent work of [23] introduced a novel packet type based approach to the D2D coded caching design with a low subpacketization size while preserving the optimal rate. As to the network security, the work of [24] proposed a D2D coded caching scheme that can avoid the information leakage, which is realized by the imperfect secret sharing technique [31]. In recent years, D2D coded caching networks have been further investigated for more practical settings, including the cases with distinct cache sizes [27], dynamic networks [28], multi-access networks [29], and wireless multi-hop networks [30].

As of now, there is scarce of work on designing efficient coded caching schemes for D2D networks, especially in realizing a low subpacketization size. Several existing ones either have a large subpacketization size or a high transmission rate, which is infeasible for the real-world applications. This paper considers the DPDA construction from the perspectives of linear algebraic and additive combinatorics, aiming to design a D2D coded caching scheme that can not only support a small subpacketization size but also achieve a low transmission rate. Our major contributions include:

- We discover a natural relationship between a linear subspace and a D2D coded caching. It is shown that a linear subspace that satisfies certain condition can be used to construct a DPDA. This enables the design of DPDA based on the existing structure of linear codes. A new class of such linear subspaces are derived through shortening the binary Reed-Muller codes [32]. By further elaborating such subspaces, a new class of DPDAs that can realize a D2D coded caching scheme with a subquadratic subpacketization size is obtained. It is shown that the proposed DPDAs can yield a large coding gain. This scheme will be characterized in *Theorem 2*.
- Inspired by the 3-term arithmetic progression (3-AP) free set of [33], we formulate a novel combinatorial structure called proper disjoint 3-AP free set. Integrating this combinatorial structure, a new framework for constructing DPDA is proposed based on the Latin square of [34]. It is shown that a DPDA with the same number of rows and columns can be developed by such arithmetic progression free sets. This property will be characterized in *Theorem 3*. Compared with the existing characterizations, the key advantage of the proposed framework is its incorporation of D2D placement and delivery strategies within the constraints of the proper disjoint 3-AP free set. Performance results show that the proposed schemes are capable of yielding a competent subpacketization size or transmission rate performance in comparison with the existing D2D coded caching schemes.

The rest of this paper is organized as follows. In Section II, we present the DPDA based coded caching network system and the relevant knowledge of binary Reed-Muller codes. The new D2D coded caching schemes are proposed in Section III. Theoretical and numerical comparisons are presented in Section IV. Our conclusions will be given in Section V.

Notations: For simplicity, the sets, vectors and arrays are denoted by curlicue letters, bolded lower-case letters and bolded upper-case letters, respectively. Let $\mathbb{Z}_q$ denote the integer ring modulo q. Let $\mathbb{Z}_q^n$ further denote a set of vectors whose elements are obtained by n-fold Cartesian product of $\mathbb{Z}_q$. Symbol $\mathbb{N}^+$ denotes the set of positive integers. We use $[m : n]$ to denote the set of consecutive integers $\{m, m+1, \ldots, n\}$. Let $d_{\rm H}(\mathbf{u}, \mathbf{v})$ denote the Hamming distance between two vectors $\mathbf{u}$ and $\mathbf{v}$, i.e., the number of coordinates that $\mathbf{u}$ and $\mathbf{v}$ differ. Let $\binom{[0:m-1]}{g}$ denote the collection of all size-g subsets of $[0 : m-1]$. Given a length-m vector $\mathbf{c}$ and a set $\mathcal{D} \subseteq [0 : m-1]$, let $\mathbf{c}|_{\mathcal{D}}$ denote a vector obtained by taking the coordinates indexed by $j \in \mathcal{D}$. Let $|\cdot|$ represent the cardinality of a set. Finally, the vectors in our examples are sometimes written as strings, e.g., $(0, 0, 0, 0)$ is written as 0000.

II. Preliminary

This section formulates the D2D coded caching network model, and briefly reviews the DPDA of [19] that can realize a D2D coded caching scheme and the relevant knowledge of binary Reed-Muller codes.

A. D2D Caching Network Model

Let us consider a centralized D2D coded caching network in Fig. 1, which includes a server having N equal-sized files. It is communicated to K users through an error free wireless

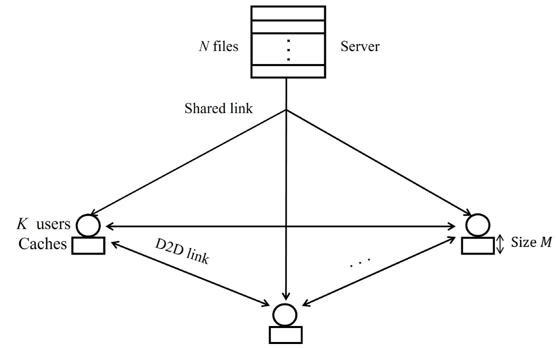

Fig. 1. D2D coded caching network.

broadcast link. Each user is equipped with a cache memory of size M files. In order to guarantee each user's demand can be satisfied, KM should be larger than N. The network users can be connected with each other through error free D2D communication links. The N files and K users are represented by two sets $\mathcal{W} = \{W_0, W_1, \ldots, W_{N-1}\}$ and $\mathcal{K} = [0 : K-1]$, respectively. The communication over this D2D network is realized by the following two phases.

- *Placement Phase*: Each file in the library is partitioned into F equal-sized packets, i.e., $W_n = \{W_n^f | f \in [0 : F-1]\}$, $n \in [0 : N-1]$. By using a well designed placement strategy, the cache of user k is filled up with some proper packets without any prior information of later user demands, where $k \in \mathcal{K}$. The packets cached by user k is denoted by $\mathcal{Z}_k$, which should be subjected to the memory constraint of $|\mathcal{Z}_k| \leq M$.
- *Delivery Phase*: The delivery phase begins when the user demands are revealed. Assume that user k requests file W_{d_k} from $\mathcal{W}$. We denote the user demand vector as $\mathbf{d} = (d_0, d_1, \ldots, d_{K-1})$, where $d_k \in [0 : N-1]$. After receiving the users' requests, a set of transmissions of size at most RF packets will be delivered over the wireless D2D communication links so that each user can reliably decode its desired file, where R is called the transmission rate. One of the important objectives in coded caching is to jointly design the cache content placement and the delivering messages so that the achievable rate for any possible demands can be minimized.

B. D2D Placement Delivery Array

The concept of DPDA was first introduced in [19], whose definition is described as follows.

Definition 1 ([19]): Given positive integers K, F, Z, and S, an $F \times K$ array $\mathbf{D} = (d_{f,k})$, where $f \in [0 : F-1]$, $k \in [0 : K-1]$, and $d_{f,k} \in [0 : S-1] \cup \{*\}$, is called a (K, F, Z, S) DPDA if the following conditions are satisfied:

B1. Each column has exactly Z "$*$" s;

B2. Each integer of $[0 : S-1]$ appears at least once in the array;

B3. For any two distinct entries d_{f_1,k_1} and d_{f_2,k_2}, $d_{f_1,k_1} = d_{f_2,k_2} = s$ is an integer only if

(a) $f_1 \neq f_2$ and $k_1 \neq k_2$, i.e., they lie in distinct rows and distinct columns;

(b) $d_{f_1,k_2} = d_{f_2,k_1} = *$, i.e., the corresponding 2×2 subarray generated by rows f_1, f_2 and columns k_1, k_2 must be in one of the following forms

$$\begin{pmatrix} s & * \\ * & s \end{pmatrix}, \begin{pmatrix} * & s \\ s & * \end{pmatrix}.$$

B4. There exists a mapping φ defined over $[0 : S-1]$ to $[0 : K-1]$ such that $d_{f,k} = s$ for any $s \in [0 : S-1]$, then $d_{f,\varphi(s)} = *$.

Algorithm 1 Coded Caching Scheme Based on DPDA [9]

Procedure Placement ($\mathbf{D}$, $\mathcal{W}$)
  Split each file $W_n \in \mathcal{W}$ into F packets as $W_n = \{W_n^f \mid f \in [0 : F-1]\}$.
  For $k \in \mathcal{K}$ **do**
    $\mathcal{Z}_k \leftarrow \{W_n^f \mid d_{f,k} = *, \forall n \in [0 : N-1]\}$;
Procedure Delivery ($\mathbf{D}, \mathcal{W}, \varphi, \mathbf{d}$)
  For $s = 0, 1, \ldots, S-1$ **do**
    User $\varphi(s)$ sends $\bigoplus_{d_{f,k}=s, f\in[0:F-1], k\in[0:K-1]} W_{d_k}^f$.

$$\mathbf{D} = \begin{pmatrix} * & 3 & 1 & * \\ 3 & * & 0 & * \\ * & 2 & * & 1 \\ 2 & * & * & 0 \end{pmatrix}, \quad \mathbf{D}' = \begin{pmatrix} 0 & * & * & * \\ * & 0 & * & * \\ * & * & 0 & * \\ * & * & * & 0 \end{pmatrix} \tag{1}$$

Note that if $\mathbf{D}$ only satisfies Conditions B1, B2 and B3, it is referred to as a PDA [6]. It can be seen that the DPDA imposes more constraints than that of the PDA. For example, the above array $\mathbf{D}$ of (1) represents a $(4, 4, 2, 4)$ DPDA, since all the conditions of *Definition 1* are satisfied. Furthermore, it is easy to check that the array $\mathbf{D}'$ is a PDA. However, it is not a DPDA due to the violation of Condition B4. Therefore, a DPDA is a PDA, but not vice versa. The DPDA based coded caching process is similar with the one realized by PDA. The main distinction between them lies in the delivery phase. The coded messages in D2D networks are transmitted from the corresponding users rather than the centralized server. This implies that the packets of coded message must be cached by at least one network user, i.e., Condition B4 needs to be further satisfied compared with the original PDA framework of [6].

Algorithm 1 was proposed to describe the DPDA based coded caching process. Given a (K, F, Z, S) DPDA $\mathbf{D}$, its row indices and column indices denote the packets and users, respectively. If $d_{f,k} = *$, user k stores the fth packet of each file in the placement phase. Condition B1 of *Definition 1* indicates that each user has a cache memory of size $M = \frac{NZ}{F}$ files. The XOR of the request packets $\bigoplus_{d_{f,k}=s, f\in[0:F-1], k\in[0:K-1]} W_{d_k}^f$ will be transmitted through the D2D communication links at the sth round of delivery. Conditions B3 and B4 of *Definition 1* ensure that each user's request packet can be obtained, since it has stored all the other packets in the coded message except the desired one. Finally, Condition B2 of *Definition 1* indicates that the number of delivering packets is equal to S, and the transmission rate will be $R = \frac{S}{F}$. Therefore, a (K, F, Z, S) DPDA $\mathbf{D}$ can realize a (K, M, N) D2D coded caching scheme with a memory ratio of

$\frac{M}{N} = \frac{Z}{F}$, an achievable rate of $R = \frac{S}{F}$, and a subpacketization size of F.

The following *Example 1* is formulated to illustrate the D2D coded caching process that is realized by a DPDA.

Example 1: The DPDA $\mathbf{D}$ of (1) can realize a $(4, 2, 4)$ D2D coded caching scheme as

- *Placement Phase*: Each file in the server is partitioned into four equal-sized packets, i.e., $W_n = \{W_n^0, W_n^1, W_n^2, W_n^3\}$, where $n \in [0:3]$. The contents cached by each user are

$$\mathcal{Z}_0 = \{W_n^0, W_n^2 \mid n \in [0:3]\};$$
$$\mathcal{Z}_1 = \{W_n^1, W_n^3 \mid n \in [0:3]\};$$
$$\mathcal{Z}_2 = \{W_n^2, W_n^3 \mid n \in [0:3]\};$$
$$\mathcal{Z}_3 = \{W_n^0, W_n^1 \mid n \in [0:3]\}.$$

- *Delivery Phase*: Suppose that the user demand vector is $\mathbf{d} = (0, 1, 2, 3)$. The messages delivered by four users are: User 0: $W_2^0 \oplus W_3^2$; User 1: $W_2^1 \oplus W_3^3$; User 2: $W_0^3 \oplus W_1^2$; User 3: $W_0^1 \oplus W_1^0$. It is easy to check that each user can reconstruct its request file. Note that there are only four coded messages delivered over the D2D communication links. The transmission rate is $R = 1$.

C. Reed-Muller Codes

Let $\mathbb{Z}_2[x_0, x_1, \ldots, x_{m-1}]$ denote a polynomial ring with m variables over binary field. Given a polynomial $h \in \mathbb{Z}_2[x_0, x_1, \ldots, x_{m-1}]$ and a vector $\mathbf{u} = (u_0, u_1, \ldots, u_{m-1}) \in \mathbb{Z}_2^m$, the evaluation of h at vector $\mathbf{u}$ is denoted as $\mathrm{E}_{\mathbf{u}}(h) = h(u_0, u_1, \ldots, u_{m-1})$. Let $\mathbf{E}(h)$ denote the evaluation vector of h whose coordinates are the evaluations of h over 2^m vectors, i.e., $\mathbf{E}(h) = (\mathrm{E}_{\mathbf{u}}(h) | \mathbf{u} \in \mathbb{Z}_2^m)$. Binary Reed-Muller code with parameters m and r can be obtained from the evaluation vectors of polynomials with m variables and degree no greater than r. In particular, the rth order binary Reed-Muller code $\mathcal{RM}(r, m)$ can be constructed by the following vectors

$$\{\mathbf{E}(h) | h \in \mathbb{Z}_2[x_0, x_1, \ldots, x_{m-1}], \deg(h) \leq r\},$$

where $\deg(h)$ denotes the algebraic degree of polynomial h. Given $\mathcal{I} \subseteq [0 : m-1]$, let $x_{\mathcal{I}} = \prod_{i \in \mathcal{I}} x_i$ denote a monomial induced by $\mathcal{I}$. Note that for any positive integer n, the equation $x^n = x$ always holds over binary field. It just needs to consider the polynomials in which the degree of each x_i is no greater than one. It can be seen that all the linear combination of the monomials in $\{x_{\mathcal{I}} | \mathcal{I} \subseteq [0 : m-1], |\mathcal{I}| \leq r\}$ will form such polynomials of degree no greater than r. Note that the above set has $\sum_{i=0}^{r} \binom{m}{i}$ monomials, and the encoding process maps the coefficients of these monomials to their corresponding evaluation vectors. This implies that $\mathcal{RM}(r, m)$ has a code length of $n = 2^m$ and a dimension of $k = \sum_{i=0}^{r} \binom{m}{i}$. Moreover, its minimum Hamming distance can be derived from binary monomial weight properties. Since $\mathcal{RM}(r, m)$ is spanned by evaluation vectors of m-variable monomials of degree no greater than r, a monomial of degree θ has an evaluation vector weight of $2^{m-\theta}$. For any non-zero codeword, let t $(t \leq r)$ denote the degree of its highest homogeneous component. Its weight is at least $2^{m-t} \geq 2^{m-r}$. A monomial of degree r can achieve this bound, thus $\mathcal{RM}(r, m)$ has a minimum Hamming distance of 2^{m-r}. The generator matrix of $\mathcal{RM}(r, m)$ can be derived by the evaluation vectors of $\{\mathbf{E}(x_{\mathcal{I}}) | \mathcal{I} \subseteq [0 : m-1], |\mathcal{I}| \leq r\}$. For some more details about Reed-Muller codes, interested readers may refer to [32]. The following *Example 2* illustrates the above generation procedure.

Example 2: Given $m = 3$ and $r = 2$, based on the evaluation vectors of monomials with 3 variables and degree no greater than 2, we have $\mathbf{E}(1) = (1,1,1,1,1,1,1,1)$; $\mathbf{E}(x_0) = (1,1,1,1,0,0,0,0)$; $\mathbf{E}(x_1) = (1,1,0,0,1,1,0,0)$; $\mathbf{E}(x_2) = (1,0,1,0,1,0,1,0)$; $\mathbf{E}(x_0x_1) = (1,1,0,0,0,0,0,0)$; $\mathbf{E}(x_0x_2) = (1,0,1,0,0,0,0,0)$; $\mathbf{E}(x_1x_2) = (1,0,0,0,1,0,0,0)$. It can be seen that these vectors can form a generator matrix of $\mathcal{RM}(2, 3)$ as

$$\mathbf{G}_0 = \begin{pmatrix} 1&1&1&1&1&1&1&1 \\ 1&1&1&1&0&0&0&0 \\ 1&1&0&0&1&1&0&0 \\ 1&0&1&0&1&0&1&0 \\ 1&1&0&0&0&0&0&0 \\ 1&0&1&0&0&0&0&0 \\ 1&0&0&0&1&0&0&0 \end{pmatrix}.$$

III. The New D2D Coded Caching Schemes

This section proposes two new theoretical frameworks of DPDA construction from the linear algebra and and additive combinatorics perspectives. The first one depends on a delicate selection of linear subspaces that satisfy certain condition. The second one is realized by strengthening a well known combinatorial structure called 3-AP free set. Using these frameworks, several small subpacketization D2D coded caching schemes are proposed with a low achievable rate.

A. Strong Linear Subspace Based Construction

A linear subspace that contains the all-ones vector as its an element is called a strong linear subspace. Let $\mathcal{A}_0$ denote a strong linear subspace over binary field with a dimension of k, minimum Hamming distance of d, and vector length of n. Let $\mathcal{A}_j$ further denote a strong linear subspace over binary field with the same dimension of k, minimum Hamming distance of $d_{\min}^{(j)} \geq d - j$, and vector length of $n - j$, where $j \in [1 : d-1]$. In order to design the later set partition, it is assumed that $k \leq n - d + 1$. Let $\mathcal{H}$ denote a vector pair set such that the Hamming distance between two vectors of each pair is greater than $n - d$, i.e.,

$$\mathcal{H} = \{(\mathbf{u}, \mathbf{v}) | d_{\mathrm{H}}(\mathbf{u}, \mathbf{v}) > n - d, \mathbf{u}, \mathbf{v} \in \mathbb{Z}_{q_0}^{n_0} \times \mathbb{Z}_{q_1}^{n_1} \times \cdots \times \mathbb{Z}_{q_{s-1}}^{n_{s-1}}\},$$

where $n = \sum_{i=0}^{s-1} n_i$. In particular, the set of vector pairs ($\mathbf{u}$, $\mathbf{v}$) such that $d_{\mathrm{H}}(\mathbf{u}, \mathbf{v}) = n - j$ is denoted by $\mathcal{H}_{d_{\mathrm{H}}=n-j}$. The vector pair set $\mathcal{H}$ is first partitioned into d disjoint subsets based on their Hamming distances, i.e., $\mathcal{H} = \bigcup_{j=0}^{d-1} \mathcal{H}_{d_{\mathrm{H}}=n-j}$. Based on the strong linear subspace $\mathcal{A}_j$, the subset $\mathcal{H}_{d_{\mathrm{H}}=n-j}$ is further partitioned into several smaller subsets. Each one can be obtained by

$$\mathcal{H}_{d_{\mathrm{H}}=n-j}^{\mathcal{S},\mathbf{j},\mathcal{A}_j,\beta} = \{(\mathbf{u}, \mathbf{v}), (\mathbf{u}', \mathbf{v}') \mid \mathbf{u}|_{\mathcal{S}} = \mathbf{v}|_{\mathcal{S}} = \mathbf{u}'|_{\mathcal{S}} = \mathbf{v}'|_{\mathcal{S}} = \mathbf{j}, (\mathbf{u}, \mathbf{v})||_{\mathbf{e}} = (\mathbf{u}', \mathbf{v}'), (\mathbf{u}, \mathbf{v}) \in \mathcal{H}_{d_{\mathrm{H}}=n-j}, (\mathbf{u}', \mathbf{v}') \in \mathcal{H}_{d_{\mathrm{H}}=n-j}\}, \quad (2)$$

where $\mathcal{S} \in \binom{[0:n-1]}{j}$ is a vector index set of size j, $\mathbf{j}$ is a vector corresponding to the same index set $\mathcal{S}$, and β is the order number of subset induced by $\mathcal{A}_j$, $\mathcal{S}$, and $\mathbf{j}$. Note that the relationship $(\mathbf{u},\mathbf{v})||_{\mathbf{e}} = (\mathbf{u}',\mathbf{v}')$ means that $(\mathbf{u}|_{[0:n-1]\setminus\mathcal{S}})|_i = (\mathbf{u}'|_{[0:n-1]\setminus\mathcal{S}})|_i$ and $(\mathbf{v}|_{[0:n-1]\setminus\mathcal{S}})|_i = (\mathbf{v}'|_{[0:n-1]\setminus\mathcal{S}})|_i$ if $e_i = 0$, and otherwise $(\mathbf{u}|_{[0:n-1]\setminus\mathcal{S}})|_i = (\mathbf{v}'|_{[0:n-1]\setminus\mathcal{S}})|_i$ and $(\mathbf{v}|_{[0:n-1]\setminus\mathcal{S}})|_i = (\mathbf{u}'|_{[0:n-1]\setminus\mathcal{S}})|_i$, where $\mathcal{S}=\emptyset$ for $j=0$, and $\mathbf{e}=(e_0,e_1,\ldots,e_{n-j-1}) \in \mathcal{A}_j$. It can be seen that these partitioned subsets are pairwise disjoint. This is because $\mathcal{A}_j$ is a strong linear subspace, and each subset can be regarded as an equivalence class. Let us present a simple example to describe how the above partitioned subsets are formulated.

Example 3: Let $\mathcal{A}_0 = \{(0,0),(1,1)\}$ and $\mathcal{A}_1 = \{(0),(1)\}$ denote two strong linear subspaces over binary field with a dimension of 1. Note that $n=2$, $d=2$, and $k=1$, we have

$$\begin{aligned}\mathcal{H} &= \{(\mathbf{u},\mathbf{v})|d_{\mathrm{H}}(\mathbf{u},\mathbf{v}) > n-d, \mathbf{u},\mathbf{v}\in\mathbb{Z}_2^2\}\\ &= \{(00,01),(00,10),(00,11),(01,00),(01,10),(01,11),\\ &\quad (10,00),(10,01),(10,11),(11,00),(11,01),(11,10)\}.\end{aligned}$$

The vector pair set $\mathcal{H}$ is first partitioned into two subsets based on the Hamming distance, i.e.,

$$\begin{aligned}\mathcal{H} &= \mathcal{H}_{d_{\mathrm{H}}=1} \cup \mathcal{H}_{d_{\mathrm{H}}=2} = \{(00,01),(00,10),(01,00),(01,11),\\ &\quad (10,00),(10,11),(11,01),(11,10)\} \cup \{(00,11),(01,10),\\ &\quad (11,00),(10,01)\}.\end{aligned}$$

Based on the strong linear subspace $\mathcal{A}_1$ and the partition rule of (2), the subset $\mathcal{H}_{d_{\mathrm{H}}=1}$ is further partitioned into four smaller subsets as

$$\begin{aligned}\mathcal{H}_{d_{\mathrm{H}}=1} &= \mathcal{H}_{d_{\mathrm{H}}=1}^{\{0\},0,\mathcal{A}_1,0} \cup \mathcal{H}_{d_{\mathrm{H}}=1}^{\{1\},0,\mathcal{A}_1,0} \cup \mathcal{H}_{d_{\mathrm{H}}=1}^{\{1\},1,\mathcal{A}_1,0} \cup\\ &\quad \mathcal{H}_{d_{\mathrm{H}}=1}^{\{0\},1,\mathcal{A}_1,0}\\ &= \{(01,00),(00,01)\} \cup \{(10,00),(00,10)\} \cup\\ &\quad \{(01,11),(11,01)\} \cup \{(10,11),(11,10)\}.\end{aligned}$$

More precisely, given $\mathcal{A}_1 = \{(0),(1)\}$, $\mathcal{S}=\{0\}$, and $\mathbf{j}=(0)$, let us select vector $(01,00)$ from $\mathcal{H}_{d_{\mathrm{H}}=1}$. If vector pairs $(\mathbf{u},\mathbf{v})$ and $(01,00)$ belong to the same subset, the first coordinate of both $\mathbf{u}$ and $\mathbf{v}$ must be 0 due to $\mathcal{S}=\{0\}$ and $\mathbf{j}=(0)$. Furthermore, given $\mathbf{e}=(1)\in\mathcal{A}_1$, it follows from the relationship $(\mathbf{u},\mathbf{v})||_{\mathbf{e}} = (\mathbf{u}',\mathbf{v}')$ that the second coordinates of $\mathbf{u}$ and $\mathbf{v}$ are 0 and 1, respectively. Therefore, we have $(\mathbf{u},\mathbf{v}) = (00,01)$. Note that the subset induced by $\mathcal{A}_1=\{(0),(1)\}$, $\mathcal{S}=\{0\}$, and $\mathbf{j}=(0)$ is unique. The order number β is equal to 0. This implies that $\mathcal{H}_{d_{\mathrm{H}}=1}^{\{0\},0,\mathcal{A}_1,0} = \{(01,00),(00,01)\}$. Similarly, the subset $\mathcal{H}_{d_{\mathrm{H}}=2}$ can also be partitioned into two subsets as

$$\begin{aligned}\mathcal{H}_{d_{\mathrm{H}}=2} &= \mathcal{H}_{d_{\mathrm{H}}=2}^{\emptyset,\mathcal{A}_0,0} \cup \mathcal{H}_{d_{\mathrm{H}}=2}^{\emptyset,\mathcal{A}_0,1}\\ &= \{(00,11),(11,00)\} \cup \{(01,10),(10,01)\}.\end{aligned}$$

For the convenience of later DPDA construction, the above partitioned subsets are represented by $\mathcal{H}_0$, $\mathcal{H}_1$, …, $\mathcal{H}_{\frac{|\mathcal{H}|}{2^k}-1}$, respectively. Applying these partitioned subsets, a new DPDA can be obtained by the following construction.

Construction 1: Given the above partitioned subsets, a $q_0^{n_0}q_1^{n_1}\cdots q_{s-1}^{n_{s-1}}(2^k-1) \times q_0^{n_0}q_1^{n_1}\cdots q_{s-1}^{n_{s-1}}$ array $\mathbf{D} = (d_{(\mathbf{u},r),\mathbf{v}})$, where $(\mathbf{u},r) \in \mathbb{Z}_{q_0}^{n_0}\times\mathbb{Z}_{q_1}^{n_1}\times\cdots\times\mathbb{Z}_{q_{s-1}}^{n_{s-1}}\times[0:2^k-2]$ and $\mathbf{v}\in\mathbb{Z}_{q_0}^{n_0}\times\mathbb{Z}_{q_1}^{n_1}\times\cdots\times\mathbb{Z}_{q_{s-1}}^{n_{s-1}}$, can be constructed with its entries defined as

$$d_{(\mathbf{u},r),\mathbf{v}} = \begin{cases}(i,\gamma), & \text{if } (u,v)\in\mathcal{H}_i \text{ and there exists an integer}\\ & \gamma \text{ such that } f(u,r)\in[\gamma(2^k-1)+1:(\gamma\\ & +1)(2^k-1)];\\ *, & \text{otherwise,}\end{cases} \tag{3}$$

where $f(\mathbf{u},r)$ denotes the occurrence number of the first coordinate i above the row indexed by $(\mathbf{u},r)$ including row $(\mathbf{u},r)$ itself. Note that $\mathbf{u}$ in row indices of $\mathbf{D}$ are arranged in the lexicographic order from top to bottom for each r.

Based on the above construction, the following result provides a class of DPDAs that can realize a D2D coded caching scheme with a subquadratic subpacketization size.

Theorem 1: Given any $n_i, s \in \mathbb{N}^+$ and distinct positive integers $q_0,q_1,\ldots,q_{s-1}$ with $q_i \geq 2$ and $\sum_{i=0}^{s-1} n_i = n$ for $i\in[0:s-1]$, there always exists a$(q_0^{n_0}q_1^{n_1}\cdots q_{s-1}^{n_{s-1}},(2^k-1)q_0^{n_0}q_1^{n_1}\cdots q_{s-1}^{n_{s-1}},(2^k-1)(q_0^{n_0}q_1^{n_1}\cdots q_{s-1}^{n_{s-1}}-\eta),q_0^{n_0}q_1^{n_1}\cdots q_{s-1}^{n_{s-1}}\eta)$DPDA which supports a$(q_0^{n_0}q_1^{n_1}\cdots q_{s-1}^{n_{s-1}}, M, N)$ D2D coded caching scheme with a memory ratio of $\frac{M}{N} = 1-\frac{\eta}{q_0^{n_0}q_1^{n_1}\cdots q_{s-1}^{n_{s-1}}}$, an achievable rate of $R=\frac{\eta}{2^k-1}$, and a subpacketization size of $(2^k-1)q_0^{n_0}q_1^{n_1}\cdots q_{s-1}^{n_{s-1}}$, where $\eta = \sum_{w=n-d+1}^{n}\sum_{\mathcal{I}\subseteq\mathcal{X},|\mathcal{I}|=w}\prod_{q_\alpha^{(\beta)}\in\mathcal{I}}(q_\alpha-1)$ and $\mathcal{X} = \{q_0^{(0)},q_0^{(1)},\ldots,q_0^{(n_0-1)},q_1^{(0)},q_1^{(1)},\ldots,q_1^{(n_1-1)},\ldots,q_{s-1}^{(0)},q_{s-1}^{(1)},\ldots,q_{s-1}^{(n_{s-1}-1)}\}$.

Proof: Without loss of generality, let us assume that $d_{(\mathbf{u},r),\mathbf{v}} = d_{(\mathbf{u}',r'),\mathbf{v}'} = (i,\gamma)$, and $\mathcal{H}_i$ is a subset derived from subspace $\mathcal{A}_j$. If $(\mathbf{u},r) = (\mathbf{u}',r')$, based on (3), it can be seen that $(\mathbf{u},\mathbf{v})\in\mathcal{H}_i$ and $(\mathbf{u},\mathbf{v}')\in\mathcal{H}_i$. This implies that $\mathbf{v}=\mathbf{v}'$ due to the partition rule of (2). Therefore, we have $(\mathbf{u},r)\neq(\mathbf{u}',r')$, i.e., each vector entry occurs in distinct rows. If $\mathbf{v}=\mathbf{v}'$, we have $(\mathbf{u},\mathbf{v})\in\mathcal{H}_i$ and $(\mathbf{u}',\mathbf{v})\in\mathcal{H}_i$. This implies that $\mathbf{u}=\mathbf{u}'$ and $r\neq r'$. Based on the above assumption, there must exist an integer γ such that $f(\mathbf{u},r)\in[\gamma(2^k-1)+1:(\gamma+1)(2^k-1)]$ and $f(\mathbf{u},r')\in[\gamma(2^k-1)+1:(\gamma+1)(2^k-1)]$. This is impossible since $|f(\mathbf{u},r)-f(\mathbf{u},r')|\geq 2^k$ and $(\gamma+1)(2^k-1)-[\gamma(2^k-1)+1] = 2^k-2$. Note that the cardinality of each partitioned subset $\mathcal{H}_i$ is 2^k. This implies that each vector entry occurs in distinct columns. Hence, Condition B3 (a) of *Definition 1* holds. Furthermore, since $(\mathbf{u},\mathbf{v})\in\mathcal{H}_i$ and $(\mathbf{u}',\mathbf{v}')\in\mathcal{H}_i$, based on (2), it can be seen that vectors $\mathbf{u}$ and $\mathbf{v}'$ agree on j coordinates. Further based on (2), there are at least $d-j$ coordinates that $\mathbf{u}$ and $\mathbf{v}'$ agree, since the Hamming distance of each two vectors in $\mathcal{A}_j$ is larger than or equal to $d-j$. This implies that the total number of coordinates that $\mathbf{u}$ and $\mathbf{v}'$ agree is larger than or equal to d, i.e., the Hamming distance between $\mathbf{u}$ and $\mathbf{v}'$ is smaller than $n-d$. Thus, we have $(\mathbf{u},\mathbf{v}')\notin\mathcal{H}$. Based on (3), we have $d_{(\mathbf{u},r),\mathbf{v}'} = *$. With a similar argument, we also have $d_{(\mathbf{u}',r'),\mathbf{v}} = *$. So Condition B3 (b) of *Definition 1* holds. It remains to show that Condition B4 of *Definition 1* holds. Suppose that $d_{(\mathbf{u},r),\mathbf{v}} = (i,\gamma)$ for $(\mathbf{u},\mathbf{v})\in\mathcal{H}_i$. Based on (3), we have $d_{(\mathbf{u},r),\mathbf{v}'} = *$, where $\mathbf{v}'$ is a vector such that $(\mathbf{u}',\mathbf{v}') \in \mathcal{H}_i$, and $\mathbf{u}'$ satisfies $f(\mathbf{u}',r') =< f(\mathbf{u},r) >_{2^k}$. Note that $f(\mathbf{u},r) = (\gamma+1)(2^k-1)+1$ and symbol $< b >_a$ denotes the least positive residue of b module a. So Condition B4 of *Definition 1* holds. Given any vector $\mathbf{v} \in \mathbb{Z}_{q_0}^{n_0} \times \mathbb{Z}_{q_1}^{n_1} \times \cdots \times \mathbb{Z}_{q_{s-1}}^{n_{s-1}}$, the number of vectors $\mathbf{u}$ such that $d_{\mathrm{H}}(\mathbf{u},\mathbf{v}) > n-d$ is $\eta = \sum_{w=n-d+1}^{n} \sum_{\mathcal{I}\subseteq\mathcal{X},|\mathcal{I}|=w} \prod_{q_\alpha^{(\beta)}\in\mathcal{I}} (q_\alpha - 1)$. Again based on (3), it can be seen that there are exactly $(2^k-1)(\mathbb{Z}_{q_0}^{n_0} \times \mathbb{Z}_{q_1}^{n_1} \times \cdots \times \mathbb{Z}_{q_{s-1}}^{n_{s-1}} - \eta)$"$*$"s in each column of $\mathbf{D}$. So Condition B1 of *Definition 1* holds. Furthermore, it can be seen that each entry of $\mathbf{D}$ appears 2^k-1 times. This implies that the number of distinct entries in $\mathbf{D}$ is $q_0^{n_0} q_1^{n_1} \cdots q_{s-1}^{n_{s-1}} \eta$. Therefore, array $\mathbf{D}(q_0^{n_0} q_1^{n_1} \cdots q_{s-1}^{n_{s-1}}, (2^k-1)q_0^{n_0} q_1^{n_1} \cdots q_{s-1}^{n_{s-1}}, (2^k-1)(q_0^{n_0} q_1^{n_1} \cdots q_{s-1}^{n_{s-1}} - \eta), q_0^{n_0} q_1^{n_1} \cdots q_{s-1}^{n_{s-1}} \eta)$ DPDA with a memory ratio of $\frac{M}{N} = 1 - \frac{\eta}{q_0^{n_0} q_1^{n_1} \cdots q_{s-1}^{n_{s-1}}}$ and a transmission rate of $R = \frac{\eta}{2^k-1}$. ■

Note that when the subspace has a dimension of $k \neq 2$, the memory ratio of the proposed scheme in *Theorem 1* will be larger than 0.5. This is because the minimum Hamming distance of nontrivial binary linear subspace cannot be greater than half of the vector length.

In particular, if the parameter $s = 1$, the following D2D coded caching scheme can be obtained. It can be considered as a special case of *Theorem 1*.

Corollary 1: Given any $n, q \in \mathbb{N}^+$ with $q \geq 2$, there always exists a $(q^n, (2^k-1)q^n, (2^k-1)(q^n-\eta), q^n\eta)$ DPDA which supports a (q^n, M, N) D2D coded caching scheme with a memory ratio of $\frac{M}{N} = 1 - \frac{\eta}{q^n}$, an achievable rate of $R = \frac{\eta}{2^k-1}$, and a subpacketization size of $(2^k-1)q^n$, where $\eta = \sum_{w=n-d+1}^{n} \binom{n}{w}(q-1)^w$.

Let us take the following *Example 4* to illustrate the realization of the scheme characterized in *Corollary 1*.

Example 4: Given two row full rank matrices

$$\mathbf{G}_0 = \begin{pmatrix} 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{pmatrix} \text{ and } \mathbf{G}_1 = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 1 & 1 \end{pmatrix},$$

we have the following two linear subspaces

$$\begin{aligned} \mathcal{A}_0 &= \{\mathbf{e} = \mathbf{h}\mathbf{G}_0 \mid \mathbf{h} \in \mathbb{Z}_2^3\} \\ &= \{(0,0,0,0),(0,0,1,1),(1,1,1,1),(0,1,0,1), \\ &\quad (1,1,0,0),(1,0,1,0),(0,1,1,0),(1,0,0,1)\}; \\ \mathcal{A}_1 &= \{\mathbf{e} = \mathbf{h}\mathbf{G}_1 \mid \mathbf{h} \in \mathbb{Z}_2^3\} = \{(0,0,0),(0,0,1),(1,1,1), \\ &\quad (0,1,0),(1,0,0),(1,0,1),(1,1,0),(0,1,1)\}. \end{aligned}$$

It can be observed that $\mathcal{A}_0$ and $\mathcal{A}_1$ are two strong linear subspaces with the same dimension of 3. Moreover, the minimum Hamming distances of them are 2 and 1, respectively. Note that $n = 4$, $d = 2$, and $n-d+1 = 3$. Given $q = 2$, we have $\mathcal{H} = \mathcal{H}_{d_{\mathrm{H}}=4} \cup \mathcal{H}_{d_{\mathrm{H}}=3}$. Based on (2) and subspace $\mathcal{A}_1$, the vector pairs in $\mathcal{H}_{d_{\mathrm{H}}=3}$ can be further partitioned into eight disjoint subsets as

$\mathcal{H}_{d_{\mathrm{H}}=3}^{\{0\},(0),\mathcal{A}_1,0} = \{(0111,0000),(0110,0001),(0101,0010),(0100,0011),(0011,0100),(0010,0101),(0001,0110),(0000,0111)\} \triangleq \mathcal{H}_0$;

$\mathcal{H}_{d_{\mathrm{H}}=3}^{\{0\},(1),\mathcal{A}_1,0} = \{(1111,1000),(1110,1001),(1101,1010),(1100,1011),(1011,1100),(1010,1101),(1001,1110),(1000,1111)\} \triangleq \mathcal{H}_1$;

$\mathcal{H}_{d_{\mathrm{H}}=3}^{\{1\},(0),\mathcal{A}_1,0} = \{(1011,0000),(1010,0001),(1001,0010),(1000,0011),(0011,1000),(0010,1001),(0001,1010),(0000,1011)\} \triangleq \mathcal{H}_2$;

$\mathcal{H}_{d_{\mathrm{H}}=3}^{\{1\},(1),\mathcal{A}_1,0} = \{(1111,0100),(1110,0101),(1101,0110),(1100,0111),(0111,1100),(0110,1101),(0101,1110),(0100,1111)\} \triangleq \mathcal{H}_3$;

$\mathcal{H}_{d_{\mathrm{H}}=3}^{\{2\},(0),\mathcal{A}_1,0} = \{(1101,0000),(1100,0001),(1001,0100),(1000,0101),(0101,1000),(0100,1001),(0001,1100),(0000,1101)\} \triangleq \mathcal{H}_4$;

$\mathcal{H}_{d_{\mathrm{H}}=3}^{\{2\},(1),\mathcal{A}_1,0} = \{(1111,0010),(1110,0011),(1011,0110),(1010,0111),(0111,1010),(0110,1011),(0011,1110),(0010,1111)\} \triangleq \mathcal{H}_5$;

$\mathcal{H}_{d_{\mathrm{H}}=3}^{\{3\},(0),\mathcal{A}_1,0} = \{(1110,0000),(1100,0010),(1010,0100),(1000,0110),(0110,1000),(0100,1010),(0010,1100),(0000,1110)\} \triangleq \mathcal{H}_6$;

$\mathcal{H}_{d_{\mathrm{H}}=3}^{\{3\},(1),\mathcal{A}_1,0} = \{(1111,0001),(1101,0011),(1011,0101),(1001,0111),(0111,1001),(0101,1011),(0011,1101),(0001,1111)\} \triangleq \mathcal{H}_7$.

Similarly, by using the linear subspace $\mathcal{A}_0$, $\mathcal{H}_{d_{\mathrm{H}}=4}$ can be partitioned into two disjoint subsets as

$\mathcal{H}_{d_{\mathrm{H}}=4}^{\emptyset,\mathcal{A}_0,0} = \{(1110,0001),(1101,0010),(1011,0100),(1000,0111),(0111,1000),(0100,1011),(0010,1101),(0001,1110)\} \triangleq \mathcal{H}_8$;

$\mathcal{H}_{d_{\mathrm{H}}=4}^{\emptyset,\mathcal{A}_0,1} = \{(1111,0000),(1100,0011),(1010,0101),(1001,0110),(0110,1001),(0101,1010),(0011,1100),(0000,1111)\} \triangleq \mathcal{H}_9$.

Based on these partitioned subsets, it follows from (3) that $d_{(0000,0),0000} = *$. This is because vector pair (0000,0000) is not included in subset $\mathcal{H}_i$, where $i \in [0:9]$. Similarly with vector pair $(0000,0111) \in \mathcal{H}_0$ and $f(0000,0) \in [0(2^3-1)+1 : (0+1)(2^3-1)]$, we can conclude that $d_{(0000,0),0111} = (0,0)$. The other entries can also be derived in the same method. As a result, a $(16,112,77,80)$ DPDA $\mathbf{D}$ can be obtained, whose first sixteen rows are presented at the end of page 7. It can be observed that this DPDA can realize a D2D coded caching scheme with the number of users $K = 16$, the memory ratio of $\frac{M}{N} = \frac{11}{16}$, the subpacketization size of $F = 112$, and the transmission rate of $R = \frac{5}{7}$.

The above construction framework depends on a class of strong linear subspaces. In fact, the linear combination of all the row vectors of generator matrix $\mathbf{G}_0$ of $\mathcal{RM}(r,m)$ can form a strong linear subspace $\mathcal{A}_0$. By deleting the last j columns of $\mathbf{G}_0$, another generator matrix $\mathbf{G}_j$ can be further obtained, which yields a strong linear subspace $\mathcal{A}_j = \{\mathbf{e} = \mathbf{h}\mathbf{G}_j \mid \mathbf{h} \in \mathbb{Z}_2^k\}$, where $j \in [1 : n-k]$. These strong linear subspaces have a property that is characterized in the following lemma.

Lemma 1: Given two integers m and r such that $0 \le r \le m$ and $m > 0$, there exists a class of strong linear subspaces $\mathcal{A}_j = \{\mathbf{e} = \mathbf{h}\mathbf{G}_j \mid \mathbf{h} \in \mathbb{Z}_2^k\}$ over binary field with the same dimension of $k = \sum_{i=0}^{r} \binom{m}{i}$ and minimum Hamming distance $d_{\min}^{(j)} \ge 2^{m-r} - j$, where $j \in [1 : n-k]$ and $n = 2^m$.

Proof: Note that the generator matrix $\mathbf{G}_0$ of $\mathcal{RM}(r,m)$ is row full rank. It can be simplified into a systematic generator matrix $\mathbf{C}_0 = (\mathbf{I}, \mathbf{B})$, where $\mathbf{I}$ is an identity matrix of order k, and $\mathbf{B}$ is a $k \times (n-k)$ matrix. It can be seen that by deleting the last j columns of $\mathbf{C}_0$, the resulting matrix $\mathbf{C}_j$ is also row full rank, where $j \in [1 : n-k]$. This implies that the dimension of the linear subspace $\mathcal{A}_j$ generated by $\mathbf{C}_j$ is k. Note that $\mathcal{A}_j = \{\mathbf{h}\mathbf{G}_j \mid \mathbf{h} \in \mathbb{Z}_2^k\} = \{\mathbf{h}\mathbf{C}_j \mid \mathbf{h} \in \mathbb{Z}_2^k\}$, where $\mathbf{G}_j$ denotes a matrix obtained by deleting the last j columns of $\mathbf{G}_0$. It can be seen that $\mathcal{A}_j$ is a strong linear subspace. Furthermore, the minimum Hamming distance between any two vectors in $\mathcal{A}_j$ is equal to the minimum Hamming weight of any nonzero vector, and the Hamming weight of any vector in $\mathcal{A}_j$ decreases by at most j through deleting the last j coordinates. Therefore, the Hamming distance between any two vectors of $\mathcal{A}_j$ is larger than or equal to $2^{m-r} - j$. ■

Note that the inequality $\sum_{i=0}^{r} \binom{m}{i} \le 2^m - 2^{m-r} + 1$ always holds for $\mathcal{RM}(r,m)$. By integrating with *Lemma 1* and *Theorem 1*, we immediately have the following result.

Theorem 2: Given any $n_i, s, m, q_0, q_1, \ldots, q_{s-1} \in \mathbb{N}^+$ and nonnegative integer r with $\sum_{i=0}^{s-1} n_i = n = 2^m$, $m \ge r$ and $q_i \ge 2$ for $i \in [0 : s-1]$, there always exists a $\left(q_0^{n_0} q_1^{n_1} \cdots q_{s-1}^{n_{s-1}}, \left(2^{\sum_{i=0}^{r}\binom{m}{i}} - 1\right) q_0^{n_0} q_1^{n_1} \cdots q_{s-1}^{n_{s-1}}, \left(2^{\sum_{i=0}^{r}\binom{m}{i}} - 1\right)(q_0^{n_0} q_1^{n_1} \cdots q_{s-1}^{n_{s-1}} - \eta), q_0^{n_0} q_1^{n_1} \cdots q_{s-1}^{n_{s-1}} \eta\right)$ DPDA which supports a $(q_0^{n_0} q_1^{n_1} \cdots q_{s-1}^{n_{s-1}}, M, N)$ D2D coded caching scheme with a memory ratio of $\frac{M}{N} = 1 - \frac{\eta}{q_0^{n_0} \cdots q_{s-1}^{n_{s-1}}}$, an achievable rate of $R = \frac{\eta}{2^{\sum_{i=0}^{r}\binom{m}{i}} - 1}$, and a subpacketization size of $\left(2^{\sum_{i=0}^{r}\binom{m}{i}} - 1\right) q_0^{n_0} q_1^{n_1} \cdots q_{s-1}^{n_{s-1}}$, where $\eta = \sum_{w=n-2^{m-r}+1}^{n} \sum_{\mathcal{I} \subseteq \mathcal{X}, |\mathcal{I}|=w} \prod_{q_\alpha^{(\beta)} \in \mathcal{I}} (q_\alpha - 1)$ and $\mathcal{X} = \{q_0^{(0)}, q_0^{(1)}, \ldots, q_0^{(n_0-1)}, q_1^{(0)}, q_1^{(1)}, \ldots, q_1^{(n_1-1)}, \ldots, q_{s-1}^{(0)}, q_{s-1}^{(1)}, \ldots, q_{s-1}^{(n_{s-1}-1)}\}$.

When we consider the parameter $s = 1$, the following D2D coded caching scheme can be further reached. Since its proof is similar with the one of *Theorem 2*, it is omitted.

Corollary 2: Given any $m, q \in \mathbb{N}^+$ and nonnegative integer r with $q \ge 2$ and $m \ge r$, there always exists a $\left(q^n, \left(2^{\sum_{i=0}^{r}\binom{m}{i}} - 1\right) q^n, \left(2^{\sum_{i=0}^{r}\binom{m}{i}} - 1\right)(q^n - \eta), q^n \eta\right)$ DPDA which supports a (q^n, M, N) D2D coded caching scheme with a memory ratio of $\frac{M}{N} = 1 - \frac{\eta}{q^n}$, an achievable rate of $R = \frac{\eta}{2^{\sum_{i=0}^{r}\binom{m}{i}} - 1}$, and a subpacketization size of $\left(2^{\sum_{i=0}^{r}\binom{m}{i}} - 1\right) q^n$, where $\eta = \sum_{w=n-2^{m-r}+1}^{n} \binom{n}{w} (q-1)^w$ and $n = 2^m$.

Theorem 1 establishes the relationship between the DPDA and the strong linear subspaces. Both the row and column indices of the constructed DPDA are generated from the corresponding vector set, while its entries are defined by the partitioned vector pair subsets. It can be seen that the subspace dimension determines the size of the partitioned subset. This implies that the transmission rate depends on the selection of strong linear subspaces. It prefers to construct a strong linear subspace that achieves a large dimension for the fixed Hamming distance. However, this is an open problem in coding theory. It is found that the binary Reed-Muller code is a linear code that contains the all-ones vector, and it has a relatively large dimension for a given Hamming distance. By shortening such code, a class of strong linear subspaces can be obtained with the same dimension. Therefore, the Reed-Muller code based scheme characterized in *Theorem 2* may yield a small transmission rate under a subquadratic subpacketization

	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
0000, 0	*	*	*	*	*	*	*	(0, 0)	*	*	*	(2, 0)	*	(4, 0)	(6, 0)	(9, 0)
0001, 0	*	*	*	*	*	*	(0, 0)	*	*	*	(2, 0)	*	(4, 0)	*	(8, 0)	(7, 0)
0010, 0	*	*	*	*	*	(0, 0)	*	*	*	(2, 0)	*	*	(6, 0)	(8, 0)	*	(5, 0)
0011, 0	*	*	*	*	(0, 0)	*	*	*	(2, 0)	*	*	*	(9, 0)	(7, 0)	(5, 0)	*
0100, 0	*	*	*	(0, 0)	*	*	*	*	*	(4, 0)	(6, 0)	(8, 0)	*	*	*	(3, 0)
0101, 0	*	*	(0, 0)	*	*	*	*	*	(4, 0)	*	(9, 0)	(7, 0)	*	*	(3, 0)	*
0110, 0	*	(0, 0)	*	*	*	*	*	*	(6, 0)	(9, 0)	*	(5, 0)	*	(3, 0)	*	*
0111, 0	(0, 1)	*	*	*	*	*	*	*	(8, 0)	(7, 0)	(5, 0)	*	(3, 0)	*	*	*
1000, 0	*	*	*	(2, 0)	*	(4, 0)	(6, 0)	(8, 0)	*	*	*	*	*	*	*	(1, 0)
1001, 0	*	*	(2, 0)	*	(4, 0)	*	(9, 0)	(7, 0)	*	*	*	*	*	*	(1, 0)	*
1010, 0	*	(2, 0)	*	*	(6, 0)	(9, 0)	*	(5, 0)	*	*	*	*	*	(1, 0)	*	*
1011, 0	(2, 1)	*	*	*	(8, 0)	(7, 0)	(5, 0)	*	*	*	*	*	(1, 0)	*	*	*
1100, 0	*	(4, 0)	(6, 0)	(9, 0)	*	*	*	(3, 0)	*	*	*	(1, 0)	*	*	*	*
1101, 0	(4, 1)	*	(8, 0)	(7, 0)	*	*	(3, 0)	*	*	*	(1, 0)	*	*	*	*	*
1110, 0	(6, 1)	(8, 1)	*	(5, 0)	*	(3, 0)	*	*	*	(1, 0)	*	*	*	*	*	*
1111, 0	(9, 1)	(7, 1)	(5, 1)	*	(3, 1)	*	*	*	(1, 1)	*	*	*	*	*	*	*

The first sixteen rows of $(16, 112, 77, 80)$ DPDA $\mathbf{D}$

size. In the proposed DPDA construction, the number of rows is $2^k - 1$ times the number of columns, where k is the dimension of strong linear subspace. Note that the number of columns is greater than $2^k - 1$. This implies that our proposed scheme can achieve an exponential gain in subpacketization size on comparison to the baseline scheme of [16]. Since the transmission rate and subpacketization size are often a tradeoff in the design of coded caching schemes, the transmission rate of the proposed scheme must be greater than the one of [16]. The detailed comparison will be appeared in Section IV-A.

B. Proper Disjoint 3-AP Free Set Based Construction

Let p denote a postive odd integer. A subset $\mathcal{B}$ of $\mathbb{Z}_p$ is called a 3-term arithmetic progression (3-AP) free set if $a + c \neq 2b$ for any three distinct integers $a, b, c \in \mathcal{B}$, i.e., it does not contain any 3-AP [33]. E.g., a subset $\mathcal{B} = \{1, 2, 4, 8\}$ of $\mathbb{Z}_9$ is a 3-AP free set, since any three distinct integers of $\mathcal{B}$ do not form a 3-AP. In this work, we strengthen this 3-AP free set, and introduce a novel combinatorial structure that was defined below.

Definition 2: Given any positive odd integer p, a set family $\mathscr{B} = \{\mathcal{B}_0, \mathcal{B}_1, \ldots, \mathcal{B}_{t-1}\}$ composed of t size-g subsets of $\mathbb{Z}_p$ is called a (p, t, g) proper disjoint 3-term arithmetic progression free set if the following constraints are satisfied:

C1. The intersection of any two distinct elements of $\mathscr{B}$ is empty;

C2. For any two distinct integers $a, b \in \mathcal{B}_i$, $\frac{a+b}{2} \notin \bigcup_{j=0}^{t-1} \mathcal{B}_j$, i.e., the half sum of any two distinct integers of $\mathcal{B}_i$ does not appear in any element of $\mathscr{B}$.

C3. For any integer $a \in \mathcal{B}_i$, $\frac{a}{2} \notin \bigcup_{j=0}^{t-1} \mathcal{B}_j$, i.e., the half of any integer of $\mathcal{B}_i$ does not appear in any element of $\mathscr{B}$.

The following *Example 5* is given to demonstrate the above definition.

Example 5: Given $p = 15$, $t = 2$, and $g = 3$, we have $\mathscr{B} = \{\mathcal{B}_0, \mathcal{B}_1\} = \{\{1, 3, 4\}, \{5, 7, 13\}\}$. This implies that $\mathcal{B}_0 \cap \mathcal{B}_1 = \emptyset$, i.e., Condition C1 of *Definition 2* holds. The half sums of any two distinct integers of $\mathcal{B}_i$ are listed as

$$\text{In } \mathcal{B}_0: \frac{1+3}{2} = 2, \frac{1+4}{2} = 10, \frac{4+3}{2} = 11;$$
$$\text{In } \mathcal{B}_1: \frac{5+7}{2} = 6, \frac{5+13}{2} = 9, \frac{13+7}{2} = 10.$$

It can be seen that $\{2, 6, 9, 10, 11\} \cap \mathcal{B}_0 = \emptyset$ and $\{2, 6, 9, 10, 11\} \cap \mathcal{B}_1 = \emptyset$. So Condition C2 of *Definition 2* holds. Finally, the half of any integer in $\mathcal{B}_i$ can be written as

$$\text{In } \mathcal{B}_0: \frac{1}{2} = 8, \frac{4}{2} = 2, \frac{3}{2} = 9;$$
$$\text{In } \mathcal{B}_1: \frac{5}{2} = 10, \frac{7}{2} = 11, \frac{13}{2} = 14,$$

which indicates that $\{2, 8, 9, 10, 11, 14\} \cap \mathcal{B}_0 = \emptyset$ and $\{2, 8, 9, 10, 11, 14\} \cap \mathcal{B}_1 = \emptyset$, i.e., Condition C3 of *Definition 2* also holds. Therefore, $\mathscr{B}$ is a $(15, 2, 3)$ proper disjoint 3-AP free set.

Note that in [34], a Latin square $\mathbf{L}$ was proposed with its entry defined as $l_{f,k} = \frac{f+k}{2}$, where $f, k \in \mathbb{Z}_p$. By appropriately adjusting its integer entries and replacing some entries with "$*$"s, an array that satisfies the DPDA constraints can be further derived. This transformation can be realized by the combinatorial structure of the proper disjoint 3-AP free set. Elaborating the use of the proper disjoint 3-AP free set, a new DPDA with the same number of rows and columns can be obtained in the following construction.

Construction 2: Given a (p, t, g) proper disjoint 3-AP free set, a $p \times p$ array $\mathbf{D} = (d_{x,y})$, where $x, y \in \mathbb{Z}_p$, can be constructed with its entries defined as

$$d_{x,y} = \begin{cases} \left(\frac{x+y}{2}, i\right), & \text{if } x - y \in \mathcal{B}_i \text{ for } i \in [0 : t-1]; \\ *, & \text{otherwise}, \end{cases} \tag{4}$$

where the arithmetic operations are performed under integer ring $\mathbb{Z}_p$.

The above construction leads to the following result, which supports a D2D coded caching scheme that has a linear subpacketization size.

Theorem 3: Given a (p, t, g) proper disjoint 3-AP free set, there always exists a $(p, p, p - tg, pt)$ DPDA which supports a (p, M, N) D2D coded caching scheme with a memory ratio of $\frac{M}{N} = 1 - \frac{tg}{p}$, an achievable rate of $R = t$, and a subpacketization size of p.

Proof: Given any $y \in \mathbb{Z}_p$, we have $\{x - y | x \in \mathbb{Z}_p\} = \mathbb{Z}_p$. This implies that each $x - y$ appears exactly once for fixed y. Note that there are t elements in $\mathscr{B}$, and each element $\mathcal{B}_i$ contains g integers. Based on (4), it can be seen that there exist tg order pairs in column y, i.e., the number of "$*$"s in column y is $p - tg$. So Condition B1 of *Definition 1* holds. For any $x \in \mathbb{Z}_p$, assume that there are two distinct integers y_1 and y_2 such that $d_{x,y_1} = d_{x,y_2} = (s, i)$, where $y_1, y_2 \in \mathbb{Z}_p$. It follows from (4) that $s = \frac{x+y_1}{2} = \frac{x+y_2}{2}$, $x - y_1 \in \mathcal{B}_i$, and $x - y_2 \in \mathcal{B}_i$. This implies that $y_1 = y_2$, which contradicts the hypothesis. So each vector entry occurs in distinct rows. Similarly, we can also show that each vector entry occurs in distinct columns. Hence, Condition B3 (a) of *Definition 1* holds. Without loss of generality, let us assume that $d_{x_1,y_1} = d_{x_2,y_2} = (s, i)$ with $x_1 \neq x_2$ and $y_1 \neq y_2$. Further based on (4), we have $s = \frac{x_1+y_1}{2} = \frac{x_2+y_2}{2}$, where $x_1 - y_1 \in \mathcal{B}_i$ and $x_2 - y_2 \in \mathcal{B}_i$. Suppose now to the contrary that $d_{x_1,y_2} \neq *$, then there exists an integer $j \in [0 : t-1]$ such that $x_1 - y_2 = c \in \mathcal{B}_j$. Let $a = x_1 - y_1$ and $b = x_2 - y_2$. We have $a - b = 2(y_2 - y_1)$ due to $x_1 + y_1 = x_2 + y_2$. This implies that $c - a = y_1 - y_2 = \frac{b-a}{2}$, i.e., $c = \frac{a+b}{2}$, where $a, b \in \mathcal{B}_i$. If $a = b$, then we have $x_1 = x_2 - y_2 + y_1$. Again based on (4), it can be seen that $s = \frac{x_1+y_1}{2} = \frac{x_2-y_2+2y_1}{2} = \frac{x_2+y_2}{2}$. This implies that $y_1 = y_2$, which contradicts the hypothesis of $y_1 \neq y_2$. Thus, we have $a \neq b$, which indicates that $c = \frac{a+b}{2} \in \mathcal{B}_j$. This is impossible since $\mathscr{B}$ is a proper disjoint 3-AP free set. Hence, we have $d_{x_1,y_2} = *$. With a similar argument, we can also show that $d_{x_2,y_1} = *$. So Condition B3 (b) of *Definition 1* holds. It remains to show that Condition B4 of *Definition 1* holds. Suppose that $d_{x,y} = (s, i)$, then we have $s = \frac{x+y}{2}$ and $x - y \in \mathcal{B}_i$. Based on the last constraint of *Definition 2*, it can be seen that $\frac{x-y}{2} \notin \mathcal{B}_j$ for $j \in [0 : t-1]$. This indicates that $x - s = \frac{x-y}{2} \notin \mathcal{B}_j$, i.e., $d_{x,s} = *$. Therefore, Condition B4 of *Definition 1* holds. Finally, let us determine the number of distinct entries of $\mathbf{D}$. Without loss of generality, let us consider the following system with two equations,

$$\begin{cases} \dfrac{x+y}{2} = s; \\ x - y = e, \end{cases}$$

where $e \in \mathcal{B}_i$. Note that given s and e, the system has a unique solution. Since the cardinality of $\mathcal{B}_i$ is equal to g, it can be seen that each entry (s, i) appears exactly g times in $\mathbf{D}$. This implies that the number of distinct entries of $\mathbf{D}$ is $\frac{tgp}{g} = tp$. Therefore, array $\mathbf{D}$ a $(p, p, p - tg, pt)$ DPDA with a memory ratio of $\frac{M}{N} = 1 - \frac{tg}{p}$ and a transmission rate of $R = t$. ■

Continued from *Example 5*, we present the following *Example 6* to demonstrate the realization of the scheme characterized in *Theorem 3*.

Example 6: Given a $(15, 2, 3)$ proper disjoint 3-AP free set of *Example 5*, based on (4), we have $d_{0,0} = *$ due to $0 - 0 = 0 \notin \mathcal{B}_0 \cup \mathcal{B}_1$. Similarly with $1 - 0 = 1 \in \mathcal{B}_0$ and $\frac{0+1}{2} = \frac{16}{2} = 8$, we can obtain $d_{0,1} = (8, 0)$. The other entry $d_{x,y}$, where $x, y \in \mathbb{Z}_{15}$, can be derived in the same manner. Consequently, the following $(15, 15, 9, 30)$ DPDA $\mathbf{D}$ can be obtained. It can provide a D2D coded caching scheme with $K = 15$ users, the memory ratio $\frac{M}{N} = \frac{3}{5}$, the subpacketization size $F = 15$, and the transmission rate $R = 2$.

The main intuition of the above example is as follows. Given a $(15, 2, 3)$ proper disjoint 3-AP free set, we first construct a diagonally cyclic Latin square $\mathbf{L}$ of order 15 with its entry defined as $l_{x,y} = \frac{x+y}{2}$, where $x, y \in \mathbb{Z}_{15}$, and replace the entry $l_{x,y}$ with "$*$" if $x - y \notin \mathcal{B}_0 \cup \mathcal{B}_1$. It can be seen that this step guarantees Condition B1 of *Definition 1*, since each column of $\mathbf{L}$ has exactly $15 - \sum_{j=0}^{1} |\mathcal{B}_j| = 9$"$*$"s. The remaining entries are then assigned with the corresponding subscript i if $x - y \in \mathcal{B}_i$. Conditions B3 (b) and B4 of *Definition 1* can be satisfied by the last two constraints of a proper disjoint 3-AP free set. Finally, since our construction is derived from a Latin square, Condition B3 (a) of *Definition 1* always holds.

TABLE I
FOUR PROPER DISJOINT 3-AP FREE SETS

Parameters	Proper disjoint 3-AP free sets
$(p,t,g)=(63,1,11)$	$\{1,3,4,9,11,12,16,25,27,48,52\}$
$(p,t,g)=(63,2,8)$	$\{\{1,3,4,9,11,12,16,25\},\{5,7,13,15,20,28,29,31\}\}$
$(p,t,g)=(63,4,6)$	$\{\{1,3,4,9,11,12\},\{5,7,13,15,20,28\},\{17,19,23,31,48,51\},\{21,27,49,53,55,60\}\}$
$(p,t,g)=(63,7,4)$	$\{\{1,3,4,9\},\{5,7,12,13\},\{15,16,19,20\},\{17,21,23,27\},\{25,28,29,48\},\{31,49,51,52\},\{53,55,60,61\}\}$

Note that the scheme characterized in *Theorem 3* depends on a (p, t, g) proper disjoint 3-AP free set. However, the corresponding parameters g and t are unknown in general, and it is challenging to provide a mathematical construction such that g can be as large as possible for the fixed p and t. Alternatively, we can formulate some kinds of proper disjoint 3-AP free sets based on the following *Algorithm 2*.

The realization of *Algorithm 2* can be partitioned into two parts. The first part in lines 1-4 aims to find a subset $\mathcal{B}'$ that satisfies the Condition C3 of *Definition 2*, which may yield a complexity of $\mathcal{O}(p)$. In each cycle, if an integer of $\mathbb{Z}'_p$ is selected to $\mathcal{B}'$, its half and double are moved to the excluded set. In the second part, $\mathcal{B}'$ is further divided into t disjoint subsets with the same size such that Condition C2 of *Definition 2* can be satisfied. This process is realized by lines 5-18 with a complexity of $\mathcal{O}(p^2)$. The remaining set is initialized by $\mathcal{B}'$. In each cycle, the integer of remaining set satisfying Condition C2 of *Definition 2* is moved to $\mathcal{B}'_i$. The cycle ends until the subset size matches the target size or the remaining set is empty. For the unallocated integers in the remaining set, lines 12-17 try to move them to the subset $\mathcal{B}'_i$ that is sorted by its size, where $i = 1, 2, \ldots, t$. Finally, line 18 constructs $\mathcal{B}_i$ with the same size. Therefore, the realization complexity of the proposed algorithm is $\mathcal{O}(p^2)$.

$$\begin{array}{c} \\ 0\\1\\2\\3\\4\\5\\6\\7\\8\\9\\10\\11\\12\\13\\14 \end{array}
\begin{array}{c} \begin{array}{ccccccccccccccc} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 \end{array} \\
\left(\begin{array}{ccccccccccccccc}
* & * & (1,1) & * & * & * & * & * & (4,1) & * & (5,1) & (13,0) & (6,0) & * & (7,0) \\
(8,0) & * & * & (2,1) & * & * & * & * & * & (5,1) & * & (6,1) & (14,0) & (7,0) & * \\
* & (9,0) & * & * & (3,1) & * & * & * & * & * & (6,1) & * & (7,1) & (0,0) & (8,0) \\
(9,0) & * & (10,0) & * & * & (4,1) & * & * & * & * & * & (7,1) & * & (8,1) & (1,0) \\
(2,0) & (10,0) & * & (11,0) & * & * & (5,1) & * & * & * & * & * & (8,1) & * & (9,1) \\
(10,1) & (3,0) & (11,0) & * & (12,0) & * & * & (6,1) & * & * & * & * & * & (9,1) & * \\
* & (11,1) & (4,0) & (12,0) & * & (13,0) & * & * & (7,1) & * & * & * & * & * & (10,1) \\
(11,1) & * & (12,1) & (5,0) & (13,0) & * & (14,0) & * & * & (8,1) & * & * & * & * & * \\
* & (12,1) & * & (13,1) & (6,0) & (14,0) & * & (0,0) & * & * & (9,1) & * & * & * & * \\
* & * & (13,1) & * & (14,1) & (7,0) & (0,0) & * & (1,0) & * & * & (10,1) & * & * & * \\
* & * & * & (14,1) & * & (0,1) & (8,0) & (1,0) & * & (2,0) & * & * & (11,1) & * & * \\
* & * & * & * & (0,1) & * & (1,1) & (9,0) & (2,0) & * & (3,0) & * & * & (12,1) & * \\
* & * & * & * & * & (1,1) & * & (2,1) & (10,0) & (3,0) & * & (4,0) & * & * & (13,1) \\
(14,1) & * & * & * & * & * & (2,1) & * & (3,1) & (11,0) & (4,0) & * & (5,0) & * & * \\
* & (0,1) & * & * & * & * & * & (3,1) & * & (4,1) & (12,0) & (5,0) & * & (6,0) & *
\end{array}\right) \end{array}.$$

$(15, 15, 9, 30)$ DPDA $\mathbf{D}$

Algorithm 2 Generating the Proper Disjoint 3-AP Free Set

Input: Positive odd integer p, subset number t;
Output: $\mathscr{B} = \{\mathcal{B}_0, \mathcal{B}_1, \ldots, \mathcal{B}_{t-1}\}$;
Initialization: Set $\mathbb{Z}'_p = \mathbb{Z}_p \setminus \{0\}$; $\mathcal{B}' = \emptyset$; $\mathcal{B}'_{\text{excluded}} = \emptyset$; $\mathcal{B}'_1 = \mathcal{B}'_2 = \ldots = \mathcal{B}'_t = \emptyset$;
Let inv2 denote the modular inverse of 2 modulo p; Calculate $\mathbb{Z}_p^{\text{double}} = (2 \times \mathbb{Z}'_p)\text{mod}p$; Calculate $\mathbb{Z}_p^{\text{half}} = (\mathbb{Z}'_p \times \text{inv2})\text{mod}p$;
for $i = 1, 2, \ldots, p-1$ **do**
  If $\mathbb{Z}'_p(i) \notin \mathcal{B}'_{\text{excluded}}$, add $\mathbb{Z}'_p(i)$ to $\mathcal{B}'$, and add $\mathbb{Z}_p^{\text{double}}(i)$ and $\mathbb{Z}_p^{\text{half}}(i)$ to $\mathcal{B}'_{\text{excluded}}$;
end for
Let $|\mathcal{B}'| = n$; Set the remaining element set $\mathcal{R} = \mathcal{B}'$; Set the target set size $Ts\left(1 : n - \lfloor \frac{n}{t} \rfloor \times t\right) = \lfloor \frac{n}{t} \rfloor + 1$;
for $i = 1, 2, \ldots, t$ **do**
  While $|\mathcal{B}'_i| < Ts(i)$ && $\mathcal{R} \neq \emptyset$
  for $j = 1, 2, \ldots, |\mathcal{R}|$ **do**
    If $((\mathcal{R}(j) + \mathcal{B}'_i) \times \text{inv2})\text{mod}p \notin \mathcal{B}'_i$, then $\mathcal{B}'_i = \mathcal{B}'_i \cup \{\mathcal{R}(j)\}$, and $\mathcal{R} = \mathcal{R} \setminus \{\mathcal{R}(j)\}$;
  end for
end for
If $\mathcal{R} \neq \emptyset$, sort the subset indices in ascending order based on its size; Denote the subset indices by $\mathcal{I}$;
for $k \in \mathcal{R}$ **do**
  for $i \in \mathcal{I}$ **do**
    If $((k + \mathcal{B}'_i) \times \text{inv2})\text{mod}p \notin \mathcal{B}'_i$, then $\mathcal{B}'_i = \mathcal{B}'_i \cup \{k\}$, and update $\mathcal{I}$;
  end for
end for
Let $g = \min\{|\mathcal{B}'_1|, |\mathcal{B}'_2|, \ldots, |\mathcal{B}'_t|\}$, and let $\mathcal{B}_{i-1}$ be an arbitrary subset of $\mathcal{B}'_i$ with a size of g;

Table I presents some proper disjoint 3-AP free sets that are generated by *Algorithm 2*.

The scheme of *Theorem 3* is derived by the (p, t, g) proper disjoint 3-AP free set $\mathscr{B}$. The size of $\mathcal{B}_i$ determines its transmission rate. *Algorithm 2* is proposed to generate a proper disjoint 3-AP free set such that g can be as large as possible for the fixed p and t. This implies that the proposed scheme of *Theorem 3* can also achieve a small transmission rate under a linear subpacketization size. Note that the DPDA derived from the proper disjoint 3-AP free set is a square array, i.e., the number of users is the same as the subpacketization size. This implies that subpacketization size of the proposed scheme is far smaller than that of [16]. However, the transmission rate is greater than that of [16].

IV. Performance Analyses of the New Scheme

This section shows the performances of the proposed D2D coded caching schemes in terms of the subpacketization size and transmission rate. Note that the memory ratios of the schemes proposed in [20] and [21] are both smaller than 0.5, while our proposed schemes are greater than 0.5. It is impossible to compare their subpacketization sizes and transmission rates with the same memory ratio. Furthermore, the scheme proposed in [22] can be regarded as a special case of the scheme characterized in *Theorem 3*. Therefore, in this section, we only compare the proposed schemes with the existing ones of [16] and [19] in Table II.

TABLE II
Summary of the Existing D2D Coded Caching Schemes

Schemes	K	$\frac{M}{N}$	R	F
Scheme in [16], any $k, t \in \mathbb{N}^+$ with $t < k$	k	$\frac{t}{k}$	$\frac{k}{t} - 1$	$t\binom{k}{t}$
Scheme in [19],any $n, p, z, t \in \mathbb{N}^+$ with $z < p$ and $t < n$	$\binom{n}{t}p^t$	$1 - (\frac{p-z}{p})^t$	$\frac{\binom{n}{t}(p-z)^t}{\binom{n}{t}\lfloor\frac{p-1}{p-z}\rfloor^t - 1}$	$(\binom{n}{t}\lfloor\frac{p-1}{p-z}\rfloor^t - 1)\lfloor\frac{p-1}{p-z}\rfloor^t p^n$
Scheme in [20], any $p \geq 2$	p^2	$\frac{1}{p}$	p	p^p
Scheme in [21], any $q, r \in \mathbb{N}^+$ with $q \geq 2r + 1$	$2q + 1$	$\frac{2r}{2q+1}$	$\frac{2q+1}{2r} - 1$	$\binom{2q+1}{2r}\frac{\binom{2r}{r} - 2^{2r}}{2^{2r}} + 2r\binom{2q+1}{2r}$
Scheme in [22], any positive integer p, and any g-subset $\mathcal{D}$ of $\mathbb{Z}_p$ with $x, y \in \mathcal{D}$, $\frac{x}{2} \notin \mathcal{D}$ and $\frac{x+y}{2} \notin \mathcal{D}$	p	$\frac{g}{p}$	1	p

A. Comparison Between the Schemes of Corollary 2, [16], and [19]

We first compare the proposed scheme with the scheme of [16]. Based on *Corollary 2*, a D2D coded caching scheme can be obtained with $K = q^n$ users, a subpacketization size of $F = \left(2^{\sum_{i=0}^{r}\binom{m}{i}} - 1\right)q^n$, the memory ratio of $\frac{M}{N} = 1 - \frac{\eta}{q^n}$, and a transmission rate of $R = \frac{\eta}{2^{\sum_{i=0}^{r}\binom{m}{i}} - 1}$. When $t = q^n - \eta$, the scheme of [16] can support the same number of users and with the same memory ratio, while its subpacketization size and transmission rate are $F' = (q^n - \eta)\binom{q^n}{\eta}$ and $R' = \frac{q^n}{q^n - \eta} - 1$, respectively. Therefore, we have

$$\frac{F'}{F} = \frac{(q^n - \eta)\binom{q^n}{\eta}}{\left(2^{\sum_{i=0}^{r}\binom{m}{i}} - 1\right)q^n}, \quad \frac{R'}{R} = \frac{2^{\sum_{i=0}^{r}\binom{m}{i}} - 1}{q^n - \eta},$$

where $\eta = \sum_{w=n-2^{m-r}+1}^{n}\binom{n}{w}(q-1)^w$. Note that the above two ratios cannot provide an intuitive conclusion. In the following we choose specific parameters for the comparison so that they can yield an exact ratio. If $q = 2$ and $r = 1$, we have $\eta = 2^{n-1} - \frac{1}{2}\binom{n}{\frac{n}{2}}$ and $n = 2^m$. The subpacketization size ratio between the schemes of [16] and *Corollary 2* becomes

$$\begin{aligned}\frac{F'}{F} &= \frac{\left(2^{n-1} + \frac{1}{2}\binom{n}{\frac{n}{2}}\right)\binom{2^n}{2^{n-1} - \frac{1}{2}\binom{n}{\frac{n}{2}}}}{(2n-1)2^n} \\ &> \frac{\left(2^{n-1} + \frac{1}{2}\binom{n}{\frac{n}{2}}\right)\binom{2^n - \binom{n}{\frac{n}{2}}}{2^{n-1} - \frac{1}{2}\binom{n}{\frac{n}{2}}}}{(2n-1)2^n} \\ &\approx \frac{\left(2^{n-1} + \frac{2^n}{\sqrt{2\pi n}}\right)2^{\left(1 - \frac{1}{\sqrt{\frac{\pi n}{2}}}\right)2^n}}{(2n-1)2^n\sqrt{\frac{\pi}{2}\left(1 - \frac{1}{\sqrt{\frac{\pi n}{2}}}\right)2^n}} = \mathcal{O}(2^{2^n}).\end{aligned}$$

TABLE III
COMPARISON BETWEEN THE SCHEME IN *COROLLARY* 2 AND THE SCHEMES OF [16] AND [19]

Schemes	Parameters	K	F	$\frac{M}{N}$	R
(m, r, q) in *Corollary 2*	$(2, 1, 2)$	16	112	0.69	0.7
(k, t) in [16]	$(16, 11)$	16	48048	0.69	0.5
(n, p, z, t) in [19]	$(2, 8, 6, 1)$	16	960	0.75	0.8
(m, r, q) in *Corollary 2*	$(3, 1, 2)$	256	3840	0.64	6.2
(k, t) in [16]	$(256, 163)$	256	6.0×10^{73}	0.64	0.6
(n, p, z, t) in [19]	$(4, 64, 41, 1)$	256	234881024	0.64	13.1
(m, r, q) in *Corollary 2*	$(3, 2, 2)$	256	32512	0.97	0.07
(k, t) in [16]	$(256, 248)$	256	1.0×10^{17}	0.97	0.03
(n, p, z, t) in [19]	$(4, 64, 62, 1)$	256	6.4×10^{10}	0.97	0.07
(m, r, q) in *Corollary 2*	$(4, 1, 2)$	65536	2031616	0.60	849.5
(k, t) in [16]	$(65536, 39321)$	65536	*Inf.*	0.60	0.7
(n, p, z, t) in [19]	$(16, 4096, 2457, 1)$	65536	9.4×10^{58}	0.60	1748.3

Note that $\binom{n}{\frac{n}{2}} \approx \frac{2^n}{\sqrt{\frac{\pi n}{2}}}$. Furthermore, the transmission rate ratio between the scheme of [16] and the scheme of *Corollary 2* can be expressed as

$$\frac{R'}{R} = \frac{2^{m+1}-1}{2^{n-1}+\frac{1}{2}\binom{n}{\frac{n}{2}}} \approx \frac{2n-1}{2^{n-1}+\frac{2^n}{\sqrt{2\pi n}}} = \mathcal{O}\left(\frac{1}{2^n}\right).$$

This implies that our proposed scheme reduces the subpacketization size substantially. Meanwhile, the transmission rate of the proposed scheme is only $\mathcal{O}(2^n)$ of that of the scheme in [16].

In order to further verify the advantages of the proposed scheme of *Corollary 2*, we compare it with the schemes of [16] and [19] in Table III. The parameters of the schemes in *Corollary 2* and [16], [19] are parameterized by (m, r, q),(k, t) and (n, p, z, t), respectively. Table III shows that in comparison with the scheme of [16], our proposed scheme achieves a smaller subpacketization size. But this is at the cost of some transmission rate. Moreover, with the same number of users and the same memory ratio, in comparison with the scheme of [19], our proposed scheme yields a lower transmission rate and also a smaller subpacketization size.

B. Comparison Between the Schemes of Theorem 3, [16], and [19]

We further compare the performance of our proposed scheme in *Theorem 3* with the schemes of [16] and [19] numerically. For the scheme of *Theorem 3*, let $(p, t, g) \in \{(177, 1, 20), (177, 2, 18), (177, 3, 17), (177, 10, 8)\}$, which is generated by *Algorithm 2*. For the scheme of [16], let $k = 177$ and $t \in [1 : 176]$. For the scheme of [19], let $p = 59$, $n = 3$, $t = 1$, and $z \in \{1\} \cup [30 : 58]$. Figs. 2 and 3 compare their transmission rate R and subpacketizaion size F against the memory ratio $\frac{M}{N}$. It can be observed that with a slightly higher transmission rate, our proposed scheme of *Theorem 3* significantly reduces the subpacketization size of the scheme in [16]. In comparison with the scheme of [19], our proposed scheme has advantages in both the subpacketization size and transmission rate. Note that the memory ratio of the proposed scheme will be greater than 0.5. This is because the proposed DPDA scheme depends on the (p, t, g) proper disjoint 3-AP free set and its size tg is smaller than $\frac{p}{2}$. However, by using the memory sharing technique, a memory ratio of less than 0.5 can be achieved.

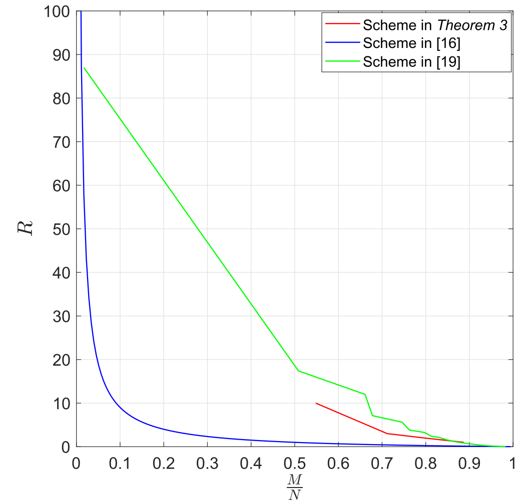

Fig. 2. Transmission rate comparison between the schemes in *Theorem 3*, and [16], [19], where $K = 177$.

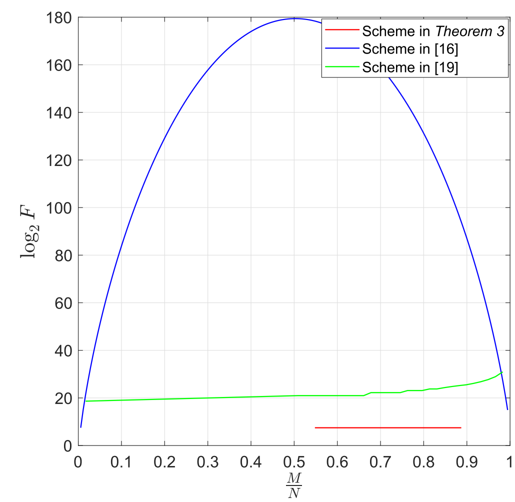

Fig. 3. Subpacketization size comparison between the schemes in *Theorem 3*, and [16], [19], where $K = 177$.

V. CONCLUSION

This paper has investigated the D2D coded caching design from the perspectives of linear algebra and additive combinatorics. It was found that designing a D2D coded caching scheme with a low subpacketization size can be transformed into problems related to the structures of linear subspace and 3-AP free set. Based on this comprehension, a new class of strong linear subspaces has been derived by shortening the binary Reed-Muller codes. Furthermore, a new combinatorial structure called proper disjoint 3-AP free set has been proposed to construct DPDAs with the same number of rows and columns. Leveraging our proposed construction frameworks, the resulting D2D coded caching schemes can exhibit a small subpacketization size while maintaining a good rate-memory tradeoff.

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] K. Wan, D. Tuninetti, and P. Piantanida, "On the optimality of uncoded cache placement," in *Proc. IEEE Inf. Theory Workshop (ITW)*, Cambridge, U.K., Sep. 2016, pp. 161–165.

[3] Q. Yu, M. A. Maddah-Ali, and A. S. Avestimehr, "The exact rate-memory tradeoff for caching with uncoded prefetching," *IEEE Trans. Inf. Theory*, vol. 64, no. 2, pp. 1281–1296, Feb. 2018.

[4] C. Tian and J. Chen, "Caching and delivery via interference elimination," *IEEE Trans. Inf. Theory*, vol. 64, no. 3, pp. 1548–1560, Mar. 2018.

[5] Q. Yu, M. A. Maddah-Ali, and A. S. Avestimehr, "Characterizing the rate-memory tradeoff in cache networks within a factor of 2," *IEEE Trans. Inf. Theory*, vol. 65, no. 1, pp. 647–663, Jan. 2019.

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

[7] H. H. S. Chittoor, M. Bhavana, and P. Krishnan, "Coded caching via projective geometry: A new low subpacketization scheme," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, France, Jul. 2019, pp. 682–686.

[8] K. Shanmugam, M. Ji, A. M. Tulino, J. Llorca, and A. G. Dimakis., "Finite-length analysis of caching-aided coded multicasting," *IEEE Trans. Inf. Theory*, vol. 62, no. 10, pp. 5524–5537, Oct. 2016.

[9] V. R. Aravind, P. K. Sarvepalli, and A. Thangaraj, "Lifting constructions of PDAs for coded caching with linear subpacketization," *IEEE Trans. Commun.*, vol. 70, no. 12, pp. 7817–7829, Dec. 2022.

[10] C. Shangguan, Y. Zhang, and G. Ge, "Centralized coded caching schemes: A hypergraph theoretical approach," *IEEE Trans. Inf. Theory*, vol. 64, no. 8, pp. 5755–5766, Aug. 2018.

[11] Q. Yan, X. Tang, Q. Chen, and M. Cheng, "Placement delivery array design through strong edge coloring of bipartite graphs," *IEEE Commun. Lett.*, vol. 22, no. 2, pp. 236–239, Feb. 2018.

[12] X. Zhong, M. Cheng, and R. Wei, "Coded caching schemes with linear subpacketizations," *IEEE Trans. Commun.*, vol. 69, no. 6, pp. 3628–3637, Jun. 2021.

[13] S. Agrawal, K. V. Sushena Sree, and P. Krishnan, "Coded caching based on combinatorial designs," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jul. 2019, pp. 1227–1231.

[14] X. Wu, M. Cheng, L. Chen, C. Li, and Z. Shi, "Design of coded caching schemes with linear subpacketizations based on injective arc coloring of regular digraphs," *IEEE Trans. Commun.*, vol. 71, no. 5, pp. 2549–2562, May 2023.

[15] K. Shanmugam, A. M. Tulino, and A. G. Dimakis, "Coded caching with linear subpacketization is possible using Ruzsa–Szeméredi graphs," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jun. 2017, pp. 1237–1241.

[16] M. Ji, G. Caire, and A. F. Molisch, "Fundamental limits of caching in wireless D2D networks," *IEEE Trans. Inf. Theory*, vol. 62, no. 2, pp. 849–869, Feb. 2016.

[17] C. Yapar, K. Wan, R. F. Schaefer, and G. Caire, "On the optimality of D2D coded caching with uncoded cache placement and one-shot delivery," *IEEE Trans. Commun.*, vol. 67, no. 12, pp. 8179–8192, Dec. 2019.

[18] A. Tebbi and C. W. Sung, "Coded caching in partially cooperative D2D communication networks," in *Proc. 9th Int. Congr. Ultra Modern Telecommun. Control Syst. Workshops (ICUMT)*, Nov. 2017, pp. 148–153.

[19] J. Wang, M. Cheng, Q. Yan, and X. Tang, "Placement delivery array design for coded caching scheme in D2D networks," *IEEE Trans. Commun.*, vol. 67, no. 5, pp. 3388–3395, May 2019.

[20] N. Woolsey, R.-R. Chen, and M. Ji, "Towards finite file packetizations in wireless device-to-device caching networks," *IEEE Trans. Commun.*, vol. 68, no. 9, pp. 5283–5298, Sep. 2020.

[21] X. Zhang, X. T. Yang, and M. Ji, "A new design framework on D2D coded caching with optimal rate and less subpacketizations," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jun. 2020, pp. 1699–1704.

[22] J. Li and Y. Chang, "New constructions of D2D placement delivery arrays," *IEEE Commun. Lett.*, vol. 27, no. 1, pp. 85–89, Jan. 2023.

[23] X. Zhang, G. Caire, and M. Ji, "Taming subpacketization without sacrificing communication: A packet type-based framework for D2D coded caching," 2026, *arXiv:2602.12220*.

[24] A. A. Zewail and A. Yener, "Device-to-device secure coded caching," *IEEE Trans. Inf. Forensics Security*, vol. 15, pp. 1513–1524, 2020.

[25] M. Ji, G. Caire, and A. F. Molisch, "Wireless device-to-device caching networks: Basic principles and system performance," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 1, pp. 176–189, Jan. 2016.

[26] M.-C. Lee, M. Ji, A. F. Molisch, and N. Sastry, "Throughput–outage analysis and evaluation of cache-aided D2D networks with measured popularity distributions," *IEEE Trans. Wireless Commun.*, vol. 18, no. 11, pp. 5316–5332, Nov. 2019.

[27] A. M. Ibrahim, A. A. Zewail, and A. Yener, "Device-to-device coded-caching with distinct cache sizes," *IEEE Trans. Commun.*, vol. 68, no. 5, pp. 2748–2762, May 2020.

[28] G. Zhu, C. Guo, T. Zhang, and Y. Shao, "Mobility-aware coded caching in D2D communication networks," *Phys. Commun.*, vol. 58, pp. 1–11, Jun. 2023.

[29] X. Wu, M. Cheng, L. Chen, R. Wu, and S. Chen, "D2D assisted coded caching design for multi-access networks," *IEEE Wireless Commun. Lett.*, vol. 13, no. 10, pp. 2702–2706, Oct. 2024.

[30] M. Ji, R.-R. Chen, G. Caire, and A. F. Molisch, "Fundamental limits of distributed caching in multihop D2D wireless networks," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jun. 2017, pp. 2950–2954.

[31] A. Shamir, "How to share a secret," *Commun. ACM*, vol. 22, no. 11, pp. 612–613, Nov. 1979.

[32] F. J. MacWilliams and N. J. A. Sloane, *The Theory of Error-Correcting Codes*. Amsterdam, The Netherlands: Elsevier, 1977.

[33] P. Frankl, R. L. Graham, and V. Rödl, "On subsets of Abelian groups with, no 3-term arithmetic progression," *J. Combinat. Theory A*, vol. 45, no. 1, pp. 157–161, May 1987.

[34] I. M. Wanless, "A partial Latin squares problem posed by Blackburn," *Bull. Inst. Combinat. Appl.*, vol. 42, pp. 76–80, Jun. 2004.

Xianzhang Wu (Senior Member, IEEE) received the B.S. degree in mathematics and applied mathematics from Minjiang University, Fuzhou, China, in 2014, the M.S. degree in applied mathematics from Fuzhou University, Fuzhou, in 2018, and the Ph.D. degree in communication and information systems from Sun Yat-sen University, Shenzhen, China, in 2023. He is currently a Lecturer with the College of Computer and Information Science, Fujian Agriculture and Forestry University, Fuzhou. His research interests include combinatorics, graph theory, caching networks, and their interactions.

Minquan Cheng (Member, IEEE) received the Ph.D. degree from the Department of Social Systems and Management, Graduate School of Systems and Information Engineering, University of Tsukuba, Tsukuba, Ibaraki, Japan, in 2012. Then, he joined Guangxi Normal University, Guilin, Guangxi, China, where he is currently a Full Professor at the School of Computer Science and Information Technology. His research interests include combinatorics, coding theory, cryptography, and their interactions.

Li Chen (Senior Member, IEEE) received the B.Sc. degree in applied physics from Jinan University, Guangzhou, China, in 2003, and the M.Sc. degree in communications and signal processing and the Ph.D. degree in communications engineering from Newcastle University, U.K., in 2004 and 2008, respectively. From 2007 to 2010, he was a Research Associate with Newcastle University. In 2010, he returned to China as a Lecturer with the School of Information Science and Technology, Sun Yat-sen University, Guangzhou. From 2011 to 2016, he was an Associate Professor and a Professor at the university. Since 2013, he has been the Associate Head of the Department of Electronic and Communication Engineering (ECE). From August 2017 to March 2020, he was the Deputy Dean of the School of ECE, SYSU. From July 2015 to June 2016, he took a sabbatical visiting both Ulm University, Germany, and the University of Notre Dame, USA. He has also visited the Institute of Network Coding, Chinese University of Hong Kong, on several occasions. He is interested in music and literature. His research interests include channel coding, in particular, and algebraic coding theory and techniques. He is a member of the IEEE Information Theory Society Board of Governors and is chairing the Conference Committee from 2022 to 2024. He founded and chairs the IEEE Information Theory Society Guangzhou Chapter, which was awarded Chapter-of-the-Year by the Society in 2021. He was awarded Chinese Information Theory Young Researcher Award by Chinese Society of Electronics in 2014. He was the TPC Co-Chair of the 2022 IEEE/CIC International Conference on Communications in China (ICCC), Foshan. He is the General Co-Chair of the 2026 IEEE International Symposium on Information Theory (ISIT), Guangzhou. He is an Associate Editor (AE) of IEEE TRANSACTIONS ON INFORMATION THEORY and was an AE of IEEE TRANSACTIONS ON COMMUNICATIONS from 2018 to 2023. He has been organizing several international conferences and workshops, including the 2018 IEEE Information Theory Workshop (ITW), Guangzhou, and the 2022 IEEE East Asian School of Information Theory (EASIT), Shenzhen, for which he was the General Co-Chair.

Congduan Li (Senior Member, IEEE) received the B.S. degree from the University of Science and Technology Beijing, China, in 2008, the M.S. degree from Northern Arizona University, AZ, USA, in 2011, and the Ph.D. degree from Drexel University, PA, USA, in 2015, respectively, all in electrical engineering. From October 2015 to August 2018, he was a Post-Doctoral Research Fellow with the Institute of Network Coding, The Chinese University of Hong Kong, and with the Department of Computer Science, City University of Hong Kong. He is currently an Associate Professor with the School of Electronics and Communication Engineering, Sun Yat-sen University, China. His research interests lie in the broad areas related to networks, such as coding, security, wireless, storage, and caching.

Shuwu Chen received the M.S. degree in radio physics from Xiamen University in 2003. He is currently a Professor with the College of Computer and Information Sciences, Fujian Agriculture and Forestry University. He is also with the Engineering Research Center of Smart Sensing and Agricultural Chip Technology, Fujian Province University. His research interests include the Internet of Things, edge computing, and AI algorithms.

Rongteng Wu (Senior Member, IEEE) received the M.S. degree from Fuzhou University, China, in 2004, and the Ph.D. degree from Tianjin University, China, in 2008. He is currently a Professor with the School of Computer and Big Data, Minjiang University, China. His current research interests include caching networks, parallel and distributed computing, cloud computing and GPU computing, and image processing.