# Low Subpacketization Coded Caching via Projective Geometry for Broadcast and D2D Networks

by

hari.hara , Prasad Krishnan

in

*IEEE Global Communications Conference*
(*GLOBECOM-2019*)

Waikoloa, HI, USA

Report No: IIIT/TR/2019/-1

Centre for Software Engineering Research Lab
International Institute of Information Technology
Hyderabad - 500 032, INDIA
December 2019

# Low Subpacketization Coded Caching via Projective Geometry for Broadcast and D2D Networks

Hari Hara Suthan Chittoor, Prasad Krishnan
Signal Processing and Communications Research Center,
International Institute of Information Technology, Hyderabad, India.
Email: {hari.hara@research., prasad.krishnan@}iiit.ac.in

***Abstract*—Coded caching was introduced as a technique of systematically exploiting locally available storage at the clients to increase the channel throughput via coded transmissions. Most known coded caching schemes in literature enable large gains in terms of the rate, however at the cost of subpacketization that is exponential in $K^{\frac{1}{r}}$ ($K$ being the number of clients, $r$ some positive integer). Building upon recent prior work for coded caching design via line graphs and finite-field projective geometries, we present a new scheme in this work which achieves a subexponential (in $K$) subpacketization of $q^{O((\log_q K)^2)}$ and rate $\Theta\left(\frac{K}{(\log_q K)^2}\right)$, for large $K$, and the cached fraction $\frac{M}{N}$ being upper bounded by a constant $\frac{2}{q^{\alpha-1}}$ (for some prime power $q$ and constant $\alpha \geq 2$) . Apart from this asymptotic improvement, we show that through some numerical comparisons that our present scheme has much lower subpacketization than previous comparable schemes, however with an increased rate and some increase in cache size required. For instance, we obtain practically relevant subpacketization levels such as $10^2 - 10^7$ for $10^2 - 10^4$ number of clients. Leveraging prior results on adapting coded caching schemes for the error-free broadcast channel to device to device (D2D) networks, we obtain a low-subpacketization scheme for D2D networks also, and give numerical comparison for the same with prior work.**

***Index Terms*—Coded caching, low subpacketization, broadcast channel, line graph, D2D networks.**

*Due to space restrictions, we give the full version of this paper in [1] with all the proofs and further results.*

## I. Introduction

Next generation wireless networks (5G and beyond) present the challenges of serving clients via channels which are not traditional point-to-point communication channels. The study of efficient high throughput communication techniques for broadcast channels, interference channels, multiple access channels, device-to-device (D2D) communication, all obtain relevance in the present wireless communication scenario. The technique of utilizing local storage (which has become quite affordable, thanks to advances in hardware design), either on the client's device or in a nearby location, for aiding communication services on a broadcast channel was introduced formally in the landmark paper [2], under the title of *Coded Caching*. In [2], it was shown that a combination of (a) carefully utilizing the local storage or *cache* available to individual clients, and (b) coded transmissions during the delivery phase, brings tremendous gains in the rate of delivery of information of a broadcast channel. Following [2], generalized cache aided communication techniques have been presented for a number of channel models [3]–[6], and in each case has shown to provide gains in the information delivery rate. Specifically, the coded caching problem for D2D networks was considered in [5], and a caching cum coded delivery scheme that was inspired from [2] was presented, which resulted similar rate advantages as [2].

The setup considered in [2] consists of an error-free broadcast channel connecting a single server with $K$ clients or receivers. The server has $N$ same-sized files, which form the library of files. Each file is divided into $F$ equal-sized subfiles ($F$ is known as the *subpacketization* parameter). Each client has a *cache* that can store $MF$ subfiles (i.e., $\frac{M}{N}$ fraction of each file). According to the scheme presented in [2] which takes place in two phases, the caches of the clients are populated by the subfiles during *the caching phase* (which occurs during off-peak time periods), and during the *demand phase* (occurring during peak-time periods) coded subfiles are transmitted to satisfy the client demands (each client demands one file in the demand phase). The *rate* $(R)$ of such a coded caching scheme is defined as the ratio of the number of bits transmitted to the size of each file, which can be calculated as

$$\text{Rate } R = \frac{\text{Number of transmissions in the delivery phase}}{\text{Number of subfiles in a file}},$$

when each transmission is of the same size as the subfiles. While the delivery scheme (and hence the rate) should ideally vary with the user demands, most works in literature starting from [2] focus on the 'worst-case' demand scenario, i..e, a scenario in which different users demand different files (under the practical constraint $K \leq N$). The rate of the scheme then corresponds to the *peak* rate. The delivery scheme in [2] consists of transmissions such that in each transmission $\gamma = 1 + \frac{MK}{N}$ clients are served. The parameter $\gamma$ is known as the *global caching gain.* The peak rate achieved is $R = \frac{K(1-\frac{M}{N})}{\gamma}$. This rate was shown to be optimal for uncoded cache placement [7] under worst-case demands. The subpacketization $F$ of the scheme in [2] is $F = \binom{K}{MK/N}$, which however becomes exponential in $K$ as $K$ grows (for constant $\frac{M}{N}$) and hence impractical even for tens of clients. In [8], an *average* rate optimal delivery scheme was presented by exploiting commonality among the user demands. In our present work, we focus on the peak rate as it represents the worst-case demands scenario. While our scheme could potentially lend itself to optimization based on commonality of demands, we leave this aspect to future work.

Since [2], several new coded caching schemes with lower subpacketization have been constructed at the cost of increase in rate, or cache requirement, or the number of users (for instance, [9] and [10]). To the best of the author's knowledge, these constructions (and others in literature for which 'large-

$K$' behaviour can be derived) have subpacketization lesser than [2] but still exponential in $K^{\frac{1}{r}}$ (for some positive integer $r$), while having larger rates compared to [2]. In particular, the scheme in [10] achieves global caching gain $\frac{MK}{N}$ with subpacketization exponential in $K$ with a much smaller exponent than [2], using a combinatorial structure called Placement Delivery Arrays (PDA). The issue of high subpacketization is carried over to the D2D problem also. An improved scheme with lower subpacketization (but higher rate) is also recently constructed for D2D networks in [11].

In this direction of research, a line graph based coded caching scheme was introduced and developed in [12], [13] to construct one of the few explicitly known *subexponential* (in $K$) subpacketization schemes. Tools from projective geometries over finite fields were used for this purpose. However the drawback of the scheme in [12] is that it needs a large cache size at clients to obtain low subpacketization. This issue was rectified in [13]. The scheme in [13] achieves a rate of $\Theta(\frac{K}{\log_q K})$ ($K$ being the number of users, $q$ is some prime power) with subexponential subpacketization $q^{O((\log_q K)^2)}$ when cached fraction is upper bounded by a small constant ($\frac{M}{N} \leq \frac{1}{q^{\alpha}}$) for some positive integer $\alpha$).

In the present work, we go further than the scheme of [13]. The tools remain the same; we use a line graph based technique combined with projective geometries over finite fields. The contributions and organization of the current work are as follows. After briefly going over the line graph coded caching approach in [12] (Section II), we present our new scheme in Section III. For large $K$ and the cached fraction $\frac{M}{N} \leq \frac{2}{q^{\alpha-1}}$ (for some constant $\alpha \geq 2$), we show that our scheme achieves rate $\Theta(\frac{K}{(\log_q K)^2})$ and subpacketization $q^{O((\log_q K)^2)}$, thus improving upon [13]. Further we also compare in Table I, by giving some numerical values to our scheme's parameters with those of [10] and [13], and show that the subpacketization achieved is several orders of magnitude lesser compared to [13] (which itself is orders of magnitude less than [10]). However the rate (equivalently, the gain) of our present scheme can be few orders of magnitude greater (equivalently, lesser) than [10] and roughly the same as [13]. Finally, in Section IV, we extend the present scheme for the error-free broadcast channel to D2D networks, utilizing a result from [14]. This results in a D2D coded caching scheme with lower subpacketization than some known schemes before. In Table II we perform a numerical comparison of the new D2D scheme with those of [5], [11].

*Notations and Terminology:* $\mathbb{Z}^+$ denotes the set of positive integers. We denote the set $\{1, \ldots, n\}$ by $[n]$ for some positive integer $n$. For sets $A, B$, the set of elements in $A$ but not in $B$ is denoted by $A \backslash B$. The finite field with $q$ elements is $\mathbb{F}_q$. The dimension of a vector space $V$ over $\mathbb{F}_q$ is given as $dim(V)$. For two subspaces $V, W$, their subspace sum is denoted by $V + W$. Note that $V + W = V \oplus W$ (the direct sum) if $V \cap W = \phi$. The span of two vectors $\mathbf{v_1}, \mathbf{v_2} \in V$, is represented as $span(\mathbf{v_1}, \mathbf{v_2})$. We give some basic definitions in graph theory. The sets $V(G), E(G)$ denote vertex set and edge set of a (simple undirected) graph $G$ respectively, where $E(G) \subseteq \{\{u, v\} : u, v \in V(G), u \neq v\}$. The square of a graph $G$ is a graph $G^2$ having $V(G^2) = V(G)$ and an edge $\{u, v\} \in E(G^2)$ if and only if either $\{u, v\} \in E(G)$ or there exists some $v_1 \in V(G)$ such that $\{u, v_1\}, \{v_1, v\} \in E(G)$. The complement of a graph $G$ is denoted as $\overline{G}$. A set $H \subseteq V(G)$ is called a clique of $G$ if every two distinct vertices in $H$ are adjacent to each other. A single vertex is also considered as a clique by definition. A clique cover of $G$ is a collection of disjoint cliques such that each vertex appears in precisely one clique.

## II. The line graph based coded caching of [12] and its relation to PDAs

Consider a coded caching system consisting of a server with files $\{W_i : i \in [N]\}$. Let $\mathcal{K}$ be any set such that $|\mathcal{K}| = K$. We shall use $\mathcal{K}$ to indicate the set of $K$ users. Let $\mathcal{F}$ be any set such that $|\mathcal{F}| = F$. The subfiles of a file $W_i$ are denoted by $W_{i,f}$ where $f \in \mathcal{F}$ and $W_{i,f}$ takes values in some Abelian group.

In [12], a line graph based framework was proposed to study the coded caching problem, which we now describe.

**Definition 1.** *(Line graph, $(c, d)$-caching line graph) [12] A graph $\mathcal{L}$ consisting of $KD$ vertices (for some $D \in \mathbb{Z}^+$), such that $\mathcal{V}(\mathcal{L}) \subseteq \mathcal{K} \times \mathcal{F}$, (for some sets $\mathcal{K}, \mathcal{F}$ such that $|\mathcal{K}| = K, |\mathcal{F}| = F$) is said to be a caching line graph (or simply, a line graph) if*

*P1: The set of vertices $\mathcal{U}_k = \{(k, f) \in V(\mathcal{L}) : f \in \mathcal{F}\}$ forms a clique of size D, for each $k \in \mathcal{K}$. (We refer to these cliques as the user cliques).*

*P2: The set of vertices $\mathcal{S}_f = \{(k, f) \in V(\mathcal{L}) : k \in \mathcal{K}\}$ forms a clique of size c (for some fixed $c \in \mathbb{Z}^+$), for each $f \in \mathcal{F}$. (We refer to these cliques as the subfile cliques).*

*P3: Each edge in $\mathcal{L}$ lies between vertices within a user clique or within a subfile clique.*

*When there is a clique cover of $\overline{\mathcal{L}^2}$ (the complement of the square of $\mathcal{L}$) consisting of disjoint $d$-sized cliques, then the line graph $\mathcal{L}$ is called as a $(c, d)$-caching line graph.*

By the above conditions P1-P3, it holds that $\bigcup_{k \in \mathcal{K}} \mathcal{U}_k = \bigcup_{f \in \mathcal{F}} \mathcal{S}_f = V(\mathcal{L})$ (these unions being disjoint). Therefore we can write $V(\mathcal{L}) = \{(k, f) \in \mathcal{K} \times \mathcal{F} : \mathcal{U}_k \cap \mathcal{S}_f \neq \phi\}$. Furthermore, it follows that $E(\mathcal{L}) = \{\{(k, f), (k', f')\} \subset V(\mathcal{L}) : k = k' \text{ or } f = f' \text{ but not both }\}$.

It was shown in [12] (refer Section IV Proposition 1 of [12]) that such a line graph $\mathcal{L}$ corresponds to a caching system in which there are $K$ users (indexed by $\mathcal{K}$) and $F$ subfiles (indexed by $\mathcal{F}$), where the $k^{th}$ user caches subfiles $\{W_{i,f} \in \mathcal{F} : \forall i \in [N]\}$ if $(k, f) \notin V(\mathcal{L})$ and does not cache them otherwise. We therefore have that each user does not cache $D$ subfiles of each file, and hence the *uncached fraction* is $1 - \frac{M}{N} = \frac{D}{F}$. Further each subfile of any file is not cached in $c$ of the $K$ users.

It is also shown in [12] that for the caching phase as defined by the line graph $\mathcal{L}$, a delivery scheme is given by a clique cover of $\overline{\mathcal{L}^2}$.

**Remark 1.** *For a $(c, d)$-caching line graph, it is shown in Theorem 2 of [12] that the parameters of the caching and*

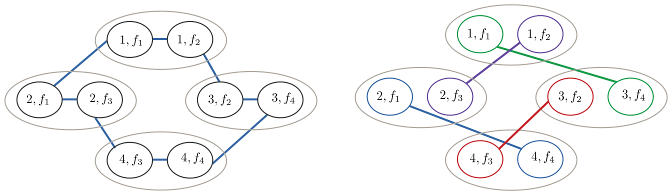

Fig. 1: The graph $\mathcal{L}$ (left) and $\overline{\mathcal{L}^2}$ (right) corresponding to the Example 1. The user cliques in $\mathcal{L}$ are placed within the ellipses, while the subfile cliques are the vertices across two user cliques connected by an edge. The cliques of $\overline{\mathcal{L}^2}$ are vertices of the same color, also connected by each edge.

*delivery scheme come out naturally, with $F = \frac{KD}{c}$ (and thus the uncached fraction being $1 - \frac{M}{N} = \frac{c}{K}$). Further the $d$-sized cliques of $\overline{\mathcal{L}^2}$ result in a delivery scheme with rate $R = \frac{c}{d}$. This is illustrated in the following example.*

**Example 1.** *Consider a coded caching system defined by the graph $\mathcal{L}$ as shown in the left of Fig. 1. This graph $\mathcal{L}$ corresponds to a coded caching setup with $K = 4$ (since it has 4 user cliques, each of size $D = 2$) and $F = 4$ (since it has 4 subfile cliques, each of size $c = 2$). Thus, there are 4 users, each user does not cache $D = 2$ subfiles (for instance, user 1 does not cache the subfiles indexed by $f_1$ and $f_2$ and caches $f_3, f_4$). Each subfile is not cached in $c = 2$ users (for instance, subfile $f_2$ is not cached in users $1, 3$). The cached fraction is $\frac{M}{N} = 1 - \frac{D}{F} = \frac{1}{2}$. The graph $\overline{\mathcal{L}^2}$ is shown on the right with 4 cliques, indicated by vertices of a same color. Corresponding to each clique of $\overline{\mathcal{L}^2}$, there is one transmission in the delivery scheme. For instance, corresponding to the clique $\{(1, f_2), (2, f_3)\}$, there is a transmission $W_{d_1,f_2} + W_{d_2,f_3}$, where $W_{d_k}$ signifies the file demanded by client $k$. Note that this enables the client 1 to decode $W_{d_1,f_2}$ and client 2 to decode $W_{d_2,f_3}$, as they each have the other subfile in their cache. Similarly, the entire set of 4 transmissions will enable decoding of all the missing subfiles at all clients.*

In the forthcoming sections, we construct a new caching line graph based scheme using projective geometry over finite fields, building on the results of [12], [13], and show that these results outperform prior known schemes in terms of the subpacketization $F$, while trading it off with some increase in the rate of the delivery scheme. Towards that end, we now recall a following structural lemma (which will be used in the next section) proved in [13] which gives the conditions under which an edge exists in $\overline{\mathcal{L}^2}$.

**Lemma 1.** *[13] Let $(k_1, f_1), (k_2, f_2) \in V(\mathcal{L})$. The edge $\{(k_1, f_1), (k_2, f_2)\} \in E(\overline{\mathcal{L}^2})$ if and only if $k_1 \neq k_2, f_1 \neq f_2$ and $(k_1, f_2) \notin V(\mathcal{L}), (k_2, f_1) \notin V(\mathcal{L})$.*

Now we recall the definition of *placement delivery array*(PDA) presented in [10].

**Definition 2** (Placement delivery array [10])**.** *For positive integers $K, F, Z$ and $S$ an $F \times K$ array $\boldsymbol{A} = [a_{j,k}], j \in [F], k \in [K]$, composed of a specific symbol "$*$" and $S$ integers $1, \cdots, S$, is called a $(K, F, Z, S)$ placement delivery array (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 $a_{j_1,k_1}$ and $a_{j_2,k_2}$ we have $a_{j_1,k_1} = a_{j_2,k_2} = s$, an integer, only if*

1. *$j_1 \neq j_2, k_1 \neq k_2$, i.e., they lie in distinct rows and distinct columns; and*
2. *$a_{j_1,k_2} = a_{j_2,k_1} = *$.*

*If each integer $s \in [S]$ occurs exactly $g$ times, $\boldsymbol{A}$ is called a regular $g - (K, F, Z, S)$ PDA, or $g$-PDA for short.*

Most known coded caching schemes in literature correspond to PDAs. We now show that any $(c, d)-$ caching line graph $(d \geq 2)$ is equivalent to a PDA.

**Lemma 2.** *$\mathcal{L}$ is a $(c, d)$-caching line graph (when $d \geq 2$ and there is a partition of $V(\mathcal{L})$ with $K$ cliques of size $D$ each) if and only if there exist a $d - (K, F = \frac{KD}{c}, Z = F - D, S = \frac{KD}{d})$ regular PDA.*

*Proof:* Available in [1]. ■

We use this lemma in Section IV to present our coded caching scheme for D2D networks.

## III. A New Scheme Based on Projective Geometry

Towards presenting our new scheme, we first review some basic concepts from projective geometry.

### A. Review of projective geometries over finite fields [15]

Let $k, q \in \mathbb{Z}^+$ such that $q$ is a prime power. Let $\mathbb{F}_q^k$ be a $k$-dim vector space over a finite field $\mathbb{F}_q$. Let '$\mathbf{0}$' represents the zero vector of $\mathbb{F}_q^k$. Consider an equivalence relation on $\mathbb{F}_q^k \backslash \{\mathbf{0}\}$ whose equivalence classes are 1-dim subspaces (without $\mathbf{0}$) of $\mathbb{F}_q^k$. The $(k-1)$-dim *projective space* over $\mathbb{F}_q$ is denoted by $PG_q(k-1)$ and is defined as the set of these equivalence classes. For $m \in [k]$, let $PG_q(k-1, m-1)$ denote the set of all $m$-dim subspaces of $\mathbb{F}_q^k$. From Chapter 3 in [15] it is known that $|PG_q(k-1, m-1)|$ is equal to the *$q$-binomial coefficient* $\begin{bmatrix} k \\ m \end{bmatrix}_q$, where $\begin{bmatrix} k \\ m \end{bmatrix}_q = \frac{(q^k-1)\dots(q^{k-m+1}-1)}{(q^m-1)\dots(q-1)}$ (where $k \geq m$). In fact, $\begin{bmatrix} k \\ m \end{bmatrix}_q$ gives the number of $m$-dim subspaces of any $k$-dim vector space over $\mathbb{F}_q$. Further, by definition, $\begin{bmatrix} k \\ 0 \end{bmatrix}_q = 1$.

Let $\mathbb{T} \triangleq \{T : T \in PG_q(k-1, 0)\}$. Let $\theta(k)$ denotes the number of distinct 1-dim subspaces of $\mathbb{F}_q^k$. Therefore $\theta(k) = |\mathbb{T}| = \begin{bmatrix} k \\ 1 \end{bmatrix}_q = \frac{q^k-1}{q-1}$.

The following lemma and corollary will be used repeatedly in this paper.

**Lemma 3.** *Let $k, a, b \in \mathbb{Z}^+$ such that $1 \leq a + b \leq k$. Consider a $k$-dim vector space $V$ over $\mathbb{F}_q$ and a fixed $a$-dim subspace $A$ of $V$. The number of distinct (un-ordered) $b$-sized sets $\{T_1, T_2, \cdots, T_b\}$ such that $T_i \in \mathbb{T}, \forall i \in [b]$ and $A \oplus T_1 \oplus T_2 \oplus \cdots \oplus T_b \in PG_q(k-1, a+b-1)$ is $\frac{\prod_{i=0}^{b-1}(\theta(k)-\theta(a+i))}{b!}$.*

*Proof:* Available in [1]. ■

**Corollary 1.** *Consider two subspaces $A, A'$ of a $k$-dim vector space $V$ over $\mathbb{F}_q$ such that $A' \subseteq A, dim(A) = a, dim(A') = a-1$. The number of distinct $T \in \mathbb{T}$ such that $A' \oplus T = A$ is* $\frac{\theta(a)-\theta(a-1)}{1!} = \frac{q^a - q^{a-1}}{q-1} = q^{a-1}$.

We now proceed to construct a caching line graph using projective geometry.

### B. A new caching line graph using projective geometry

Consider $k, m, t \in \mathbb{Z}^+$ such that $m+t+2 \leq k$. Consider a $k$-dim vector space $\mathbb{F}_q^k$. Let $W$ be a fixed $(t-1)$-dim subspace of $\mathbb{F}_q^k$. Consider the following sets of subspaces, where each such subspace contains $W$.

$$\mathbb{V} \triangleq \{V \in PG_q(k-1, t-1) : W \subseteq V\}.$$
$$\mathbb{R} \triangleq \{R \in PG_q(k-1, t) : W \subseteq R\}.$$
$$\mathbb{S} \triangleq \{S \in PG_q(k-1, m+t-1) : W \subseteq S\}.$$
$$\mathbb{U} \triangleq \{U \in PG_q(k-1, m+t+1) : W \subseteq U\}.$$

Now, consider the following sets, which are used to present our line graph and the corresponding coded caching scheme.

$$\mathbb{X} \triangleq \{\{V_1, V_2\} : V_1, V_2 \in \mathbb{V}, V_1 + V_2 \in \mathbb{R}\}. \quad (1)$$
$$\mathbb{Y} \triangleq \left\{\{V_1, V_2, \cdots, V_{m+1}\} : \forall V_i \in \mathbb{V}, \sum_{i=1}^{m+1} V_i \in \mathbb{S}\right\}. \quad (2)$$
$$\mathbb{Z} \triangleq \left\{\{V_1, V_2, \cdots, V_{m+3}\} : \forall V_i \in \mathbb{V}, \sum_{i=1}^{m+3} V_i \in \mathbb{U}\right\}. \quad (3)$$

To construct a caching line graph $\mathcal{L}$, we need to satisfy the conditions P1-P3 in Definition 1. Following the notations in Section II, let $\mathcal{K} = \mathbb{X}$ and $\mathcal{F} = \mathbb{Y}$. We construct $\mathcal{L}$ systematically by first initializing $\mathcal{L}$ by its user-cliques. The user-cliques are indexed by $X \in \mathbb{X}$. For each $X \in \mathbb{X}$ create the vertices corresponding to the user-clique indexed by $X$ as $C_X \triangleq \left\{(X, Y) : Y \in \mathbb{Y}, \sum_{V_i \in X} V_i + \sum_{V_i \in Y} V_i \in \mathbb{U}\right\}$. Thus, $V(\mathcal{L}) \triangleq \bigcup_{X \in \mathbb{X}} C_X$. Now, for each $Y \in \mathbb{Y}$ we construct the subfile clique of $\mathcal{L}$ associated with $Y$ as $C_Y \triangleq \left\{(X, Y) : X \in \mathbb{X}, \sum_{V_i \in X} V_i + \sum_{V_i \in Y} V_i \in \mathbb{U}\right\}$. We thus see that $V(\mathcal{L}) = \bigcup_{Y \in \mathbb{Y}} C_Y$. Now, if we show that the user cliques (and equivalently, subfile cliques) are of the same size each, then the properties P1-P3 will be satisfied by $\mathcal{L}$. By invoking the notations from Section II, we have $K = |\mathbb{X}|$ (number of user-cliques), and subpacketization $F = |\mathbb{Y}|$ (the number of subfile cliques).

We now find the values of $K, F$, the size of user clique $|C_X|$ and the size of subfile clique $|C_Y|$.

**Lemma 4.**

$$K = |\mathbb{X}| = \frac{q}{2} \begin{bmatrix} k-t+1 \\ 1 \end{bmatrix}_q \begin{bmatrix} k-t \\ 1 \end{bmatrix}_q.$$

$$F = |\mathbb{Y}| = \begin{bmatrix} k-t+1 \\ m+1 \end{bmatrix}_q \frac{\prod_{i=0}^{m} (q^{m+1} - q^i)}{(m+1)!(q-1)^{(m+1)}}.$$

$$|C_X| = \frac{1}{(m+1)!} \; q^{\frac{(m+1)(m+4)}{2}} \prod_{i=1}^{m+1} \begin{bmatrix} k-t-i \\ 1 \end{bmatrix}_q.$$

$$|C_Y| = \frac{q^{(2m+3)}}{2} \begin{bmatrix} k-m-t \\ 1 \end{bmatrix}_q \begin{bmatrix} k-m-t-1 \\ 1 \end{bmatrix}_q.$$

*for any $X \in \mathbb{X}$, $Y \in \mathbb{Y}$.*

*Proof:* Available in [1]. ■

Note that by Lemma 4, we have the size of the subfile cliques of $\mathcal{L}$ as $|C_Y|$ (for any $Y \in \mathbb{Y}$), and this is same for each $Y$. Similarly the user-cliques all have the same size $|C_X|$. Hence $\mathcal{L}$ satisfies properties P1-P3. We now show that $\overline{\mathcal{L}^2}$ has a clique cover with $d$-sized disjoint cliques for some $d$. Therefore $\mathcal{L}$ is in fact a $(c = |C_Y|, d)$-caching line graph, giving rise to the main result in this section which is Theorem 1.

### C. Delivery Scheme from a clique cover of $\overline{\mathcal{L}^2}$

We first describe a clique of $\overline{\mathcal{L}^2}$ and show that such equal-sized cliques partition $V(\mathcal{L}) = V(\overline{\mathcal{L}^2})$. This will suffice to show the delivery scheme as per Theorem 2 of [12] (summarized in Remark 1 in Section II in this work).

We now present a clique of size $\binom{m+3}{2}$ in $\overline{\mathcal{L}^2}$ (where $\binom{a}{b}$ represents binomial coefficient). Recall the definition of $\mathbb{Z}$ from (3).

**Lemma 5.** *Consider $Z = \{V_1, V_2, \cdots, V_{m+3}\} \in \mathbb{Z}$. Then $C_Z = \{(\{V_i, V_j\}, Z \setminus \{V_i, V_j\}), \forall V_i, V_j \in Z, i \neq j\} \subseteq V(\overline{\mathcal{L}^2})$ is a clique in $\overline{\mathcal{L}^2}$.*

*Proof:* Available in [1]. ■

Now we show that the cliques $\{C_Z : Z \in \mathbb{Z}\}$ partition $V(\overline{\mathcal{L}^2})$.

**Lemma 6.** $\bigcup_{Z \in \mathbb{Z}} C_Z = V(\mathcal{L}) = V(\overline{\mathcal{L}^2})$*, where this union is a disjoint union (the cliques $C_Z$ are as defined in Lemma 5).*

*Proof:* Available in [1]. ■

Finally we present our coded caching scheme using the caching line graph constructed above.

**Theorem 1.** *The caching line graph $\mathcal{L}$ constructed above is a* $\left(c = \frac{q^{(2m+3)}}{2} \begin{bmatrix} k-m-t \\ 1 \end{bmatrix}_q \begin{bmatrix} k-m-t-1 \\ 1 \end{bmatrix}_q, d = \binom{m+3}{2}\right)$*-caching line graph and defines a coded caching scheme with*

$$K = \frac{q}{2} \begin{bmatrix} k-t+1 \\ 1 \end{bmatrix}_q \begin{bmatrix} k-t \\ 1 \end{bmatrix}_q,$$

$$F = \begin{bmatrix} k-t+1 \\ m+1 \end{bmatrix}_q \frac{\prod_{i=0}^{m} (q^{m+1} - q^i)}{(m+1)!(q-1)^{m+1}},$$

$$\frac{M}{N} = 1 - q^{2(m+1)} \frac{\begin{bmatrix} k-m-t \\ 1 \end{bmatrix}_q \begin{bmatrix} k-m-t-1 \\ 1 \end{bmatrix}_q}{\begin{bmatrix} k-t+1 \\ 1 \end{bmatrix}_q \begin{bmatrix} k-t \\ 1 \end{bmatrix}_q},$$

$$R = \frac{q^{(2m+3)}}{(m+2)(m+3)} \begin{bmatrix} k-m-t \\ 1 \end{bmatrix}_q \begin{bmatrix} k-m-t-1 \\ 1 \end{bmatrix}_q.$$

*Proof:* Available in [1]. ■

We now present Algorithm 1, which presents the caching and delivery scheme developed in this section. For the given $K', M', N'$ select the appropriate parameters $k, m, t, q$ which give $K, \frac{M}{N}, F, R$ such that $(K - K')$ and $(\frac{M'}{N'} - \frac{M}{N})$ are non negative and as small as possible (we treat the extra users $K - K'$ as dummy users). Now construct a $(c, d)$-caching line graph $(\mathcal{L})$ as mentioned in Section III-B and find $\mathbb{X}$ (user indices), $\mathbb{Y}$ (subfile indices), $\mathbb{Z}$ (indices of cliques of $\overline{\mathcal{L}^2}$, equivalently indices of transmissions) by using (1),(2),(3).

**Algorithm 1** Coded caching scheme proposed in Theorem 1

```
procedure PLACEMENT PHASE
    for each $i \in [N']$ do
        Split $W_i$ into $\{W_{i,Y} : Y \in \mathbb{Y}\}$.
    end for
    for each $X \in \mathbb{X}$ do
        user $X$ caches the subfiles $W_{i,Y}, \forall i \in [N'], \forall Y \in \mathbb{Y}$ such that $(X, Y) \notin V(\mathcal{L})$.
    end for
end procedure
procedure DELIVERY PHASE( demand of user $X$ is represented as $W_{d_X}, \forall X \in \mathbb{X}$)
    for each $Z = \{V_1, V_2, \cdots, V_{m+3}\} \in \mathbb{Z}$ do
        Server transmits $\sum_{\{V_i,V_j\} \subset Z} W_{d_{\{V_i,V_j\}}, Z \backslash \{V_i,V_j\}}$.
    end for
end procedure
```

In the full version of our paper [1], we present an asymptotic analysis of the behaviour of the subpacketization $F$ and the rate $R$ of our scheme as $K$ grows large, with $\frac{M}{N}$ bounded from above by a constant ($\frac{M}{N} \leq \frac{2}{q^{\alpha}-1}$, for some integer $\alpha \geq 2$). We show that $F = q^{O((\log_q K)^2)}$ and $R = \Theta(\frac{K}{(\log_q K)^2})$. Thus our scheme has a subexponential subpacketization (in $K$), however the rate is not constant as $K \to \infty$ (unlike [2], [10]).

Finally in Table I, we compare numerically the scheme in Theorem 1 with the scheme in [13] and [10] for some choices of $K, U = 1 - \frac{M}{N}, F$ and $\gamma$ (the global caching gain, i.e., $\frac{K(1-\frac{M}{N})}{R}$, where $R$ is the rate achieved by the scheme).

We label the parameters of our scheme in Theorem 1 as $K_1, U_1, F_1, \gamma_1$ where $\gamma_1 = d$. The parameters of the scheme presented in [13] are labeled as $K_2, U_2, F_2, \gamma_2$( for explicit expressions, the reader is referred to [13]). Parameters of the scheme presented in [10] are $K_3 = q'(m' + 1), U_3 = 1 - \frac{1}{q'}, F_3 = (q')^{(m')}, \gamma_3 = \frac{K(1-\frac{M}{N})}{q'-1}$ where $q'(\geq 2), m' \in \mathbb{Z}^+$. As subpacketization can be very large, we approximate it to the nearest positive power of 10.

We see from the table that the proposed scheme performs much better than [13], [10] in terms of the subpacketization (note that the subpacketization of [10] is less than that of the original scheme, [2]). In particular, the subpacketization obtained by our scheme is much lower than that of [13] (which is much lesser than [10]), even for thousands of clients it remains reasonable and practical. The global caching gain however is close to that of [13], and few orders of magnitude smaller than that in [10] (thus the rate of our scheme is comparable to [13] and larger than [10]). This indicates that our scheme can be implemented in the practical broadcast coded caching networks.

## IV. APPLICATION TO D2D NETWORKS

We now adapt our new coded caching scheme to a scheme for D2D networks by utilizing a result of [14]. First we describe the D2D network model as in [5] briefly. In contrast to the conventional coded caching setup, the central server is absent in D2D network. In a D2D coded caching network there is a library of $N^{\mathcal{D}}$ files, $K^{\mathcal{D}}$ users each equipped with a cache memory that can store $M^{\mathcal{D}}$ number of files. Each file is divided into $F^{\mathcal{D}}$ (subpacketization) number of equal sized subfiles. All users are connected by a bus link. During one time slot any one of the users can transmit and other users can receive (without error). The D2D coded caching system works in two phases. During the caching phase, the cache memory of each user is populated with contents available at the library (with the constraint that each cache can store $M^{\mathcal{D}}$ files). During the transmission phase each user demands any one of the files available in the library. The demand of each user is revealed to all other users. Every user ($l \in [K^{\mathcal{D}}]$) makes a multicast transmission of rate $r_l$ (the ratio of number of coded subfiles transmitted by $l$ to the subpacketization $F^{\mathcal{D}}$), in its dedicated time slot to all other users using the bus link. From these multicast transmissions and cache contents each user decodes its demanded file. The rate of the D2D coded caching system is defined as $R^{\mathcal{D}} = \sum_{l=1}^{K^{\mathcal{D}}} r_l$.

Similar to the conventional coded caching, the practical D2D coded caching systems demand low subpacketization schemes with lower rates.

In [14] it was shown that for any $g$-PDA with $g \geq 2$ there exists a corresponding D2D coded caching scheme (for the explicit construction, the reader is referred to Theorem 1 in [14]).

**Lemma 7** (Corollary 1 in [14])**.** *For a given $g-(K, F, Z, S)$ regular PDA with $g \geq 2$, there exists a scheme for a D2D network with $K^{\mathcal{D}} = K$ users and cached fraction $\frac{M^{\mathcal{D}}}{N^{\mathcal{D}}} = \frac{Z}{F}$, achieving the rate $R^{\mathcal{D}} = \frac{g}{g-1}\frac{S}{F}$, with subpacketization level $F^{\mathcal{D}} = (g-1)F$. (Here the parameters with superscript $\mathcal{D}$ represents the parameters of the D2D coded caching scheme)*

By Lemma 2 it is easy to see that the $(c, d)$-caching line graph developed in Section III-B corresponds to a $d - (K, F, F - D, \frac{KD}{d})$ regular PDA. Now, by applying Lemma 7, we can get the corresponding D2D coded caching scheme which is presented in the Theorem 2 (the proof follows from Lemma 2 and Lemma 7).

**Theorem 2.** *The caching line graph given in Section III-B corresponds to a D2D coded caching scheme with*

$$K^{\mathcal{D}} = \frac{q}{2} \begin{bmatrix} k - t + 1 \\ 1 \end{bmatrix}_q \begin{bmatrix} k - t \\ 1 \end{bmatrix}_q .$$

$$F^{\mathcal{D}} = \frac{(m+1)(m+4)}{2} \begin{bmatrix} k - t + 1 \\ m + 1 \end{bmatrix}_q \frac{\prod_{i=0}^{m} (q^{m+1} - q^i)}{(m+1)!(q-1)^{m+1}}.$$

| $K_1$ | $K_2$ | $K_3$ | $U_1$ | $U_2$ | $U_3$ | $F_1$ | $F_2$ | $F_3$ | $\gamma_1$ | $\gamma_2$ | $\gamma_3$ |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | [13] | [10] | | [13] | [10] | | [13] | [10] | | [13] | [10] |
| 8001 | 8191 | 8008 | 0.93 | 0.94 | 0.93 | 8001 | $10^{29}$ | inf | 6 | 10 | 572 |
| 8001 | 8191 | 8001 | 0.67 | 0.75 | 0.67 | $10^7$ | $10^{35}$ | inf | 15 | 12 | $10^3$ |
| 780 | 781 | 780 | 0.62 | 0.80 | 0.67 | 780 | $10^{10}$ | $10^{123}$ | 6 | 5 | 260 |
| 465 | 511 | 468 | 0.72 | 0.75 | 0.75 | 465 | $10^{15}$ | $10^{69}$ | 6 | 8 | 117 |
| 105 | 127 | 104 | 0.46 | 0.50 | 0.50 | 105 | $10^9$ | $10^{15}$ | 6 | 7 | 52 |

TABLE I: Comparison of Coded caching schemes presented in [13], [10] with this work. (inf represents $> 10^{307}$).

| $K_1^{\mathcal{D}}$ | $K_2^{\mathcal{D}}$ | $K_3^{\mathcal{D}}$ | $U_1^{\mathcal{D}}$ | $U_2^{\mathcal{D}}$ | $U_3^{\mathcal{D}}$ | $F_1^{\mathcal{D}}$ | $F_2^{\mathcal{D}}$ | $F_3^{\mathcal{D}}$ | $R_1^{\mathcal{D}}$ | $R_2^{\mathcal{D}}$ | $R_3^{\mathcal{D}}$ |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | [11] | [5] | | [11] | [5] | | [11] | [5] | | [11] | [5] |
| 8001 | 8001 | 8001 | 0.93 | 0.98 | 0.93 | 40005 | $10^{174}$ | inf | 1488 | 89.44 | 13.28 |
| 7260 | 7260 | 7260 | 0.87 | 0.98 | 0.87 | 36300 | $10^{164}$ | inf | 1263 | 85.20 | 6.70 |
| 1953 | 1953 | 1953 | 0.86 | 0.97 | 0.86 | 9765 | $10^{72}$ | inf | 336 | 44.19 | 6.15 |
| 780 | 780 | 780 | 0.62 | 0.96 | 0.62 | 3900 | $10^{40}$ | $10^{225}$ | 97.2 | 27.92 | 1.65 |
| 465 | 465 | 465 | 0.72 | 0.95 | 0.72 | 2325 | $10^{28}$ | $10^{119}$ | 67.2 | 21.56 | 2.60 |
| 105 | 105 | 105 | 0.46 | 0.90 | 0.46 | 525 | $10^{10}$ | $10^{32}$ | 9.6 | 10.25 | 0.84 |

TABLE II: Comparison of D2D Coded caching schemes presented in [11], [5] with this work. (inf represents $> 10^{307}$).

$$\frac{M^{\mathcal{D}}}{N^{\mathcal{D}}} = 1 - q^{2(m+1)} \frac{\begin{bmatrix} k-m-t \\ 1 \end{bmatrix}_q \begin{bmatrix} k-m-t-1 \\ 1 \end{bmatrix}_q}{\begin{bmatrix} k-t+1 \\ 1 \end{bmatrix}_q \begin{bmatrix} k-t \\ 1 \end{bmatrix}_q}.$$

$$R^{\mathcal{D}} = \frac{q^{(2m+3)}}{(m+1)(m+4)} \begin{bmatrix} k-m-t \\ 1 \end{bmatrix}_q \begin{bmatrix} k-m-t-1 \\ 1 \end{bmatrix}_q.$$

In [1], we obtain the asymptotics of this D2D scheme for large $K$ and $\frac{M}{N}$ bounded by a constant as $F^{\mathcal{D}} = q^{O((\log_q K^{\mathcal{D}})^2)}$ and $R^{\mathcal{D}} = \Theta(\frac{K^{\mathcal{D}}}{(\log_q K^{\mathcal{D}})^2})$ (where $K^{\mathcal{D}}$ is the number of users in the D2D coded caching scheme).

In Table II, we compare the D2D coded caching scheme presented in Theorem 2 with that of [5], [11]. As far as possible, we choose corresponding values for $K^{\mathcal{D}}, U^{\mathcal{D}} = 1 - \frac{M^{\mathcal{D}}}{N^{\mathcal{D}}}$. The parameters corresponding to Theorem 2 are labelled as $K_1^{\mathcal{D}}, U_1^{\mathcal{D}}, F_1^{\mathcal{D}}, R_1^{\mathcal{D}}$. The parameters corresponding to the scheme of [11] are $K_2^{\mathcal{D}}, U_2^{\mathcal{D}} = 1 - \frac{1}{y_1}, F_2^{\mathcal{D}} = y_1^{y_1}, R_2^{\mathcal{D}} = y_1$ where $y_1 = \sqrt{K^{\mathcal{D}}}$. In this scheme we only have freedom to choose $K_2^{\mathcal{D}}$, all other parameters depend on $K_2^{\mathcal{D}}$ (because of this we are unable to even approximately match $U_2^{\mathcal{D}}$ with $U_1^{\mathcal{D}}$ of our scheme).

The parameters corresponding to the scheme of [5] are $K_3^{\mathcal{D}}, U_3^{\mathcal{D}} = 1 - \frac{M^{\mathcal{D}}}{N^{\mathcal{D}}}, F_3^{\mathcal{D}} = y_2\binom{K^{\mathcal{D}}}{y_2}, R_3^{\mathcal{D}} = \frac{N^{\mathcal{D}}}{M^{\mathcal{D}}} - 1$ where $y_2 = \lfloor \frac{M^{\mathcal{D}} K^{\mathcal{D}}}{N^{\mathcal{D}}} \rfloor$. From Table II it is clear that the scheme presented in Theorem 2 performs much better than the schemes of [5], [11] in terms of subpacketization but with higher rate, which indicates that our scheme can be implemented in the practical D2D coded caching networks.

## ACKNOWLEDGMENT

This work was supported partly by the Early Career Research Award (ECR/2016/000447) and the MATRICS grant (MTR/2017/000827 ) from Science and Engineering Research Board (Govt.of India) to Prasad Krishnan. Hari Hara Suthan Chittoor was supported by Visvesvaraya PhD Scheme for Electronics and IT (Govt.of India).

## REFERENCES

[1] H. H. S. Chittoor and P. Krishnan, “Low subpacketization coded caching via projective geometry for broadcast and d2d networks,” *CoRR*, vol. abs/1902.08041, 2019, https://arxiv.org/abs/1902.08041.

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

[3] N. Naderializadeh, M. A. Maddah-Ali, and A. S. Avestimehr, “Fundamental limits of cache-aided interference management,” *IEEE Transactions on Information Theory*, vol. 63, no. 5, pp. 3092–3107, May 2017.

[4] M. A. Maddah-Ali and U. Niesen, “Decentralized coded caching attains order-optimal memory-rate tradeoff,” *IEEE/ACM Transactions on Networking*, vol. 23, no. 4, pp. 1029–1040, Aug 2015.

[5] M. Ji, G. Caire, and A. F. Molisch, “Fundamental limits of caching in wireless d2d networks,” *IEEE Transactions on Information Theory*, vol. 62, no. 2, pp. 849–869, Feb 2016.

[6] M. Ji, M. F. Wong, A. M. Tulino, J. Llorca, G. Caire, M. Effros, and M. Langberg, “On the fundamental limits of caching in combination networks,” in *2015 IEEE 16th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, June 2015, pp. 695–699.

[7] K. Wan, D. Tuninetti, and P. Piantanida, “On the optimality of uncoded cache placement,” in *2016 IEEE Information Theory Workshop (ITW)*, Sep. 2016, pp. 161–165.

[8] Q. Yu, M. A. Maddah-Ali, and A. S. Avestimehr, “The exact rate-memory tradeoff for caching with uncoded prefetching,” *IEEE Transactions on Information Theory*, vol. 64, no. 2, pp. 1281–1296, Feb 2018.

[9] L. Tang and A. Ramamoorthy, “Coded caching schemes with reduced subpacketization from linear block codes,” *IEEE Transactions on Information Theory*, vol. 64, no. 4, pp. 3099–3120, April 2018.

[10] 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, Sep. 2017.

[11] N. Woolsey, R. Chen, and M. Ji, “Coded caching in wireless device-to-device networks using a hypercube approach,” in *2018 IEEE International Conference on Communications Workshops (ICC Workshops)*, May 2018, pp. 1–6.

[12] P. Krishnan, “Coded caching via line graphs of bipartite graphs,” in *2018 IEEE Information Theory Workshop (ITW)*, Nov 2018, pp. 1–5.

[13] H. H. S. Chittoor, B. M, and P. Krishnan, “Coded caching via projective geometry: A new low subpacketization scheme,” in *2019 IEEE International Symposium on Information Theory (ISIT)*, July. 2019, pp. 682–686.

[14] Q. Yan, X. Tang, and Q. Chen, “Placement delivery array and its applications,” in *2018 IEEE Information Theory Workshop (ITW)*, Nov 2018, pp. 1–5.

[15] J. Hirschfeld, *Projective Geometries Over Finite Fields. Oxford Mathematical Monographs.* Oxford University Press New York, 1998.